[](https://ida.interchain.io/)

[Interchain Developer Academy](https://ida.interchain.io/)/[Interchain Developer Academy](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)



Search

[Interchain Developer Academy](https://ida.interchain.io/)[Interchain Developer Academy](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

Search



Filters

Interchain Developer Academy

[](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Week 0 - Getting Started](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Getting Started](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Blockchain 101](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Blockchain History](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Public and Managed Blockchains](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Consensus in Distributed Networks](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Cryptography](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Self-Assessment Quiz](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Go Introduction - First Steps](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Go Basics](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Go Interfaces](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Control Structures in Go](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Arrays and Slices in Go](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Standard Packages in Go](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Concurrency in Go](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Good-To-Know Dev Terms](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Docker Introduction](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Week 1 - Introduction to the Interchain](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Introduction to the Interchain](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Blockchain Technology and the Interchain](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[The Interchain Ecosystem](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Getting ATOM and Staking It](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[A Blockchain App Architecture](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Accounts](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Transactions](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Messages](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Modules](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Protobuf](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Multistore and Keepers](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[BaseApp](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Queries](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Events](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Context](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Testing](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Relaying with IBC](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Interchain Security](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Bridges](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Migrations](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Week 1 Quiz](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Week 2 - First Steps](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[First Steps](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Setup Your Work Environment](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Run a Node, API, and CLI](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Ignite CLI](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Exercise - Make a Checkers Blockchain](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Store Object](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Create Custom Messages](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Create and Save a Game Properly](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Add a Way to Make a Move](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Emit Game Information](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Record the Game Winner](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Week 2 Exercise](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Week 3 - Introduction to IBC and CosmJS](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Introduction to IBC and CosmJS](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[What is IBC?](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[IBC/TAO - Connections (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[IBC/TAO - Channels (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[IBC/TAO - Clients (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[IBC Token Transfer](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Interchain Accounts (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[IBC Middleware (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Create a Custom IBC Middleware (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Integrate IBC Middleware Into a Chain (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[IBC Tooling](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[What is CosmJS?](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Your First CosmJS Actions](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Compose Complex Transactions](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Learn to Integrate Keplr](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Create Custom CosmJS Interfaces](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Week 4 - Ignite CLI and IBC Advanced](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Ignite CLI and IBC Advanced](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Keep an Up-To-Date Game Deadline](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Keep Track Of How Many Moves Have Been Played](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Put Your Games in Order](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Auto-Expiring Games](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Let Players Set a Wager](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Handle wager payments](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Integration tests](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Incentivize Players](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Help Find a Correct Move](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Play With Cross-Chain Tokens](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Understand IBC Denoms](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Go Relayer](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Hermes Relayer](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Week 5 - CosmJS Advanced](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[CosmJS Advanced](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Create Custom Objects](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Create Custom Messages](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Get an External GUI](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Integrate CosmJS and Keplr](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Backend Script for Game Indexing](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Week 6 - IBC Deep Dive](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[IBC Deep Dive](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[IBC Application Developer Introduction](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Make a Module IBC-Enabled](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Adding Packet and Acknowledgment Data](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Extend the Checkers Game With a Leaderboard](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Create a Leaderboard Chain](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Week 7 - From Code to MVP to Production and Migrations](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[From Code to MVP to Production and Migrations](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Run in Production](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Prepare the Software to Run](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Prepare a Validator and Keys](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Prepare Where the Node Starts](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Prepare and Connect to Other Nodes](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Configure, Run, and Set Up a Service](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Prepare and Do Migrations](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Simulate Production in Docker](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Tally Player Info After Production](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Add a Leaderboard as a Module](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Migrate the Leaderboard Module After Production](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Simulate a Migration in Docker](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Final Exam](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[What's Next?](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

[Continue Your Interchain Journey](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html)

Docs Version Switcher

On this page

[What to expect](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#what-to-expect)

[Prepare checkers executables](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#prepare-checkers-executables)

[Blockchain elements](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#blockchain-elements)

[Prepare the Cosmovisor executable](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#prepare-the-cosmovisor-executable)

[Docker Compose elements](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#docker-compose-elements)

[Run all the elements](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#run-all-the-elements)

[Add games](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#add-games)

[Prepare the upgrade proposals](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#prepare-the-upgrade-proposals)

[Send the first upgrade proposal](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#send-the-first-upgrade-proposal)

[Vote on the first proposal](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#vote-on-the-first-proposal)

[Refill your cup](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#refill-your-cup)

[Send the second upgrade proposal](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#send-the-second-upgrade-proposal)

[The first live upgrade](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#the-first-live-upgrade)

[The second live upgrade](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#the-second-live-upgrade)

[What about stop and restart?](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#what-about-stop-and-restart)

[#Copy link](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#simulate-a-migration-in-docker) **Simulate a Migration in Docker**



Make sure you have all you need before proceeding:

* You understand the concepts of [migrations](https://ida.interchain.io/academy/2-cosmos-concepts/16-migrations.html), [production](https://ida.interchain.io/tutorials/9-path-to-prod/), and [migrations in production](https://ida.interchain.io/tutorials/9-path-to-prod/7-migration.html).
* Docker is installed and you [understand it](https://ida.interchain.io/tutorials/5-docker-intro/).
* You have the checkers blockchain codebase up to the migrations. If not, follow the [previous steps](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/4-migration-leaderboard.html) or check out the [relevant version (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/tree/leaderboard-migration).



In this section, you will:

* Prepare Docker elements.
* Deal with data migrations.
* Upgrade your blockchain in production with Cosmovisor.
* Compose everything in one orchestrated ensemble.
* Test it.

In previous sections, you have:

* Simulated a production setup with the help of Docker Compose.
* Learned how to implement an in-place migration.

This section is about:

* Running an in-place migration...
* ...in a simulated production setup...
* ...with the help of Docker Compose.

You will reuse the nodes, validators, and sentries created for Alice, Bob, and Carol in a [previous section](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/1-run-prod-docker.html).

[#Copy link](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#what-to-expect) What to expect

In this section, you will accomplish the following steps:

1. Build the checkers v2 software.
2. Build the checkers v1.1 software.
3. Build the checkers v1 software.
4. Build Cosmovisor and set it up for two consecutive known upgrades on all nodes.
5. Launch everything.
6. Create the first upgrade governance proposal, v1tov1\_1.
7. Have the first proposal pass.
8. Repeat with the second upgrade governance proposal, v1\_1tov2.
9. Observe the first migration take place.
10. Have the second proposal pass.
11. Observe the second migration take place.
12. Stop everything and start it again safely in v2.

In a real production situation, node operators would wait for a named upgrade governance proposal to be on the ballot (if not wait for it to be approved) before they went to the trouble of setting up Cosmovisor; therefore point 6 would happen some time before points 2 and 4. However, because we know that the proposal will go through, we will use the above order in the interest of time.

Also, both proposals will be created before the first upgrade, and the second proposal with be *passed* after the first upgrade. That is in the interest of the exercise, so as to see the second proposal cross the first upgrade in its pending state.

[#Copy link](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#prepare-checkers-executables) Prepare checkers executables

At this stage, your checkers code already has the migration elements. It is therefore in its v2 configuration. Create the corresponding Docker image:



Copy

$ docker build . \

-f prod-sim/Dockerfile-checkersd-alpine \

-t checkersd\_i:v2-alpine

To build v1 and v1.1, you first need to check out the code before the migration code steps were added. Depending on your branch names, this would be:

**Build v1.1**

**Build v1**



Copy

$ git checkout player-info-migration

$ docker build . \

-f prod-sim/Dockerfile-checkersd-alpine \

-t checkersd\_i:v1.1-alpine

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/tree/player-info-migration)

Copy

$ git checkout run-prod

$ docker build . \

-f prod-sim/Dockerfile-checkersd-alpine \

-t checkersd\_i:v1-alpine

[View source](https://github.com/cosmos/b9-checkers-academy-draft/tree/run-prod)

Do not forget to come back to your v2 branch. For instance, with:



Copy

$ git checkout leaderboard-migration

[#Copy link](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#blockchain-elements) Blockchain elements

Your genesis elements should still be in v1 as they were created in the run-prod branch. You can confirm this by verifying that there are no player infos and no leaderboards in the [checkers genesis store (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/blob/run-prod/prod-sim/node-carol/config/genesis.json#L85).

As you did in the [migration section](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/2-migration-info.html), you need to reduce the voting period from 2 days to 10 minutes to make the exercise bearable:



Copy

$ jq -j '.app\_state.gov.voting\_params.voting\_period = "600s"' prod-sim/desk-alice/config/genesis.json > prod-sim/desk-alice/config/genesis-2.json && mv prod-sim/desk-alice/config/genesis-2.json prod-sim/desk-alice/config/genesis.json

$ jq -j '.app\_state.gov.voting\_params.voting\_period = "600s"' prod-sim/desk-bob/config/genesis.json > prod-sim/desk-bob/config/genesis-2.json && mv prod-sim/desk-bob/config/genesis-2.json prod-sim/desk-bob/config/genesis.json

$ jq -j '.app\_state.gov.voting\_params.voting\_period = "600s"' prod-sim/sentry-alice/config/genesis.json > prod-sim/sentry-alice/config/genesis-2.json && mv prod-sim/sentry-alice/config/genesis-2.json prod-sim/sentry-alice/config/genesis.json

$ jq -j '.app\_state.gov.voting\_params.voting\_period = "600s"' prod-sim/sentry-bob/config/genesis.json > prod-sim/sentry-bob/config/genesis-2.json && mv prod-sim/sentry-bob/config/genesis-2.json prod-sim/sentry-bob/config/genesis.json

$ jq -j '.app\_state.gov.voting\_params.voting\_period = "600s"' prod-sim/val-alice/config/genesis.json > prod-sim/val-alice/config/genesis-2.json && mv prod-sim/val-alice/config/genesis-2.json prod-sim/val-alice/config/genesis.json

$ jq -j '.app\_state.gov.voting\_params.voting\_period = "600s"' prod-sim/val-bob/config/genesis.json > prod-sim/val-bob/config/genesis-2.json && mv prod-sim/val-bob/config/genesis-2.json prod-sim/val-bob/config/genesis.json

$ jq -j '.app\_state.gov.voting\_params.voting\_period = "600s"' prod-sim/node-carol/config/genesis.json > prod-sim/node-carol/config/genesis-2.json && mv prod-sim/node-carol/config/genesis-2.json prod-sim/node-carol/config/genesis.json

The names of the upgrade proposals will be v1tov2\_1 and v1\_1tov2. The names are important, as they are defined in the code, and Cosmovisor uses them to determine which executable to run.

[#Copy link](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#prepare-the-cosmovisor-executable) Prepare the Cosmovisor executable

Because the project in its current state uses Cosmos SDK v0.45.4, to avoid any surprise you will prepare Cosmovisor at the [v0.45.4 (opens new window)↗](https://docs.cosmos.network/v0.45/run-node/cosmovisor.html) version too.

You can describe the steps in a new Dockerfile prod-sim/Dockerfile-cosmovisor-alpine, described here logically (before a recap lower down):

1. You need to build Cosmovisor from its code:



Copy

FROM --platform=linux golang:1.18.7-alpine AS builder

ENV COSMOS\_VERSION=v0.45.4

RUN apk update

RUN apk add make git

WORKDIR /root

RUN git clone --depth 1 --branch ${COSMOS\_VERSION} https://github.com/cosmos/cosmos-sdk.git

WORKDIR /root/cosmos-sdk/cosmovisor

RUN make cosmovisor

FROM --platform=linux alpine

ENV LOCAL=/usr/local

COPY --from=builder /root/cosmos-sdk/cosmovisor/cosmovisor ${LOCAL}/bin/cosmovisor

prod-sim /

Dockerfile-cosmovisor-alpine

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/migration-prod/prod-sim/Dockerfile-cosmovisor-alpine" \l "L1-L26" \t "_blank)

1. Cosmovisor is instructed via [environment variables (opens new window)↗](https://docs.cosmos.network/v0.45/run-node/cosmovisor.html#command-line-arguments-and-environment-variables). In the eventual containers, the /root/.checkers folder comes from a volume mount, so to avoid any conflict it is better not to put the cosmovisor folder directly inside it. Instead pick /root/.checkers-upgrade:



Copy

...

FROM --platform=linux alpine

ENV LOCAL=/usr/local

+ ENV DAEMON\_HOME=/root/.checkers-upgrade

+ ENV DAEMON\_NAME=checkersd

+ ENV DAEMON\_ALLOW\_DOWNLOAD\_BINARIES=false

+ ENV DAEMON\_RESTART\_AFTER\_UPGRADE=true

...

prod-sim /

Dockerfile-cosmovisor-alpine

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/migration-prod/prod-sim/Dockerfile-cosmovisor-alpine" \l "L21-L24" \t "_blank)

1. With the folder decided, you can introduce all three checkers executables. They can be conveniently taken from their respective Docker images:



Copy

+ FROM --platform=linux checkersd\_i:v1-alpine AS v1

+ FROM --platform=linux checkersd\_i:v1.1-alpine AS v1.1

+ FROM --platform=linux checkersd\_i:v2-alpine AS v2

FROM --platform=linux alpine

...

COPY --from=builder /root/cosmos-sdk/cosmovisor/cosmovisor ${LOCAL}/bin/cosmovisor

+ COPY --from=v1 /usr/local/bin/checkersd $DAEMON\_HOME/cosmovisor/genesis/bin/checkersd

+ COPY --from=v1.1 /usr/local/bin/checkersd $DAEMON\_HOME/cosmovisor/upgrades/v1tov1\_1/bin/checkersd

+ COPY --from=v2 /usr/local/bin/checkersd $DAEMON\_HOME/cosmovisor/upgrades/v1tov2/bin/checkersd

...

prod-sim /

Dockerfile-cosmovisor-alpine

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/migration-prod/prod-sim/Dockerfile-cosmovisor-alpine" \l "L15-L29" \t "_blank)

Checkers starts at v1, therefore the v1 executable goes into .../genesis. We know that:

* + The executable of the eventual upgrade named v1tov1\_1 is the v1.1 one.
  + The executable of the eventual upgrade named v1\_1tov2 is the v2 one.

Note also the decision to use /usr/local explicitly, as this is knowledge that is kept in a separate Docker image.

1. Now make Cosmovisor start by default:



Copy

COPY --from=v2 /usr/local/bin/checkersd $DAEMON\_HOME/cosmovisor/upgrades/v1tov2/bin/checkersd

+ ENTRYPOINT [ "cosmovisor" ]

prod-sim /

Dockerfile-cosmovisor-alpine

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/migration-prod/prod-sim/Dockerfile-cosmovisor-alpine" \l "L31" \t "_blank)

When you put all this together, you get:



Copy

FROM --platform=linux golang:1.18.7-alpine AS builder

ENV COSMOS\_VERSION=v0.45.4

RUN apk update

RUN apk add make git

WORKDIR /root

RUN git clone --depth 1 --branch ${COSMOS\_VERSION} https://github.com/cosmos/cosmos-sdk.git

WORKDIR /root/cosmos-sdk/cosmovisor

RUN make cosmovisor

FROM --platform=linux checkersd\_i:v1-alpine AS v1

FROM --platform=linux checkersd\_i:v1.1-alpine AS v1.1

FROM --platform=linux checkersd\_i:v2-alpine AS v2

FROM --platform=linux alpine

ENV LOCAL=/usr/local

ENV DAEMON\_HOME=/root/.checkers-upgrade

ENV DAEMON\_NAME=checkersd

ENV DAEMON\_ALLOW\_DOWNLOAD\_BINARIES=false

ENV DAEMON\_RESTART\_AFTER\_UPGRADE=true

COPY --from=builder /root/cosmos-sdk/cosmovisor/cosmovisor ${LOCAL}/bin/cosmovisor

COPY --from=v1 /usr/local/bin/checkersd $DAEMON\_HOME/cosmovisor/genesis/bin/checkersd

COPY --from=v1.1 /usr/local/bin/checkersd $DAEMON\_HOME/cosmovisor/upgrades/v1tov1\_1/bin/checkersd

COPY --from=v2 /usr/local/bin/checkersd $DAEMON\_HOME/cosmovisor/upgrades/v1\_1tov2/bin/checkersd

ENTRYPOINT [ "cosmovisor" ]

prod-sim /

Dockerfile-cosmovisor-alpine

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/migration-prod/prod-sim/Dockerfile-cosmovisor-alpine" \t "_blank)

Now you can create the Cosmovisor Docker image with a meaningful tag:



Copy

$ docker build . \

-f prod-sim/Dockerfile-cosmovisor-alpine \

-t cosmovisor\_i:v1tov2-alpine

[#Copy link](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#docker-compose-elements) Docker Compose elements

With the executables and the blockchain elements ready, you can now define the *production setup*. You already defined one in the previous [run checkers in prod section](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/1-run-prod-docker.html). In this new setup, the only things that change are the Docker images you call: cosmovisor\_i instead of checkersd\_i. Even the start command does not need to change.

To avoid rewriting everything, you can declare a Docker Compose [extension (opens new window)↗](https://docs.docker.com/compose/extends/) in a new file prod-sim/docker-compose-cosmovisor.yml. Each checkersd type of service is extended, and in the end is replaced, with a new image:



Copy

version: "3.7"

services:

val-alice:

extends:

file: docker-compose.yml

service: val-alice

image: cosmovisor\_i:v1tov2-alpine

sentry-alice:

extends:

file: docker-compose.yml

service: sentry-alice

image: cosmovisor\_i:v1tov2-alpine

val-bob:

extends:

file: docker-compose.yml

service: val-bob

image: cosmovisor\_i:v1tov2-alpine

sentry-bob:

extends:

file: docker-compose.yml

service: sentry-bob

image: cosmovisor\_i:v1tov2-alpine

node-carol:

extends:

file: docker-compose.yml

service: node-carol

image: cosmovisor\_i:v1tov2-alpine

prod-sim /

docker-compose-cosmovisor.yml

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/migration-prod/prod-sim/docker-compose-cosmovisor.yml" \t "_blank)



docker-compose-cosmovisor.yml's val-alice extends docker-compose.yml's val-alice, while keeping the same name. In effect this overwrites val-alice, instead of starting another validator working on the same shared prod-sim/val-alice folder.

[#Copy link](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#run-all-the-elements) Run all the elements

Now you can run everything and confirm that all services start:



Copy

$ docker compose \

--file prod-sim/docker-compose.yml \

--file prod-sim/docker-compose-cosmovisor.yml \

--project-name checkers-prod up \

--detach

In fact, at this stage there is no difference from the previous prod setup, which was what you now call v1. Blocks are being created.

Confirm you are on v1:

* One way to confirm is to use checkers v1.1 to query Carol's node for the player infos:



Copy

$ docker run --rm -it \

--network checkers-prod\_net-public \

checkersd\_i:v1.1-alpine \

query checkers list-player-info \

--node "tcp://node-carol:26657"

It returns:



Copy

Error: rpc error: code = Unknown desc = unknown query path: unknown request

This confirms that the anwering node does not know about such structure.

* Another way to confirm is to see to which folder Cosmovisor's current folder is symbolically linking:



Copy

$ docker exec -it node-carol \

ls -l /root/.checkers-upgrade/cosmovisor/current

This should return:



Copy

... /root/.checkers-upgrade/cosmovisor/current -> /root/.checkers-upgrade/cosmovisor/genesis

Again, this confirms that it is running v1, as found in .../genesis.

[#Copy link](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#add-games) Add games

You will now need Alice and Bob's addresses, so take them from the keyrings found on their respective *desktops*.

**Alice**

**Bob**



Copy

$ alice=$(echo password | docker run --rm -i \

-v $(pwd)/prod-sim/desk-alice:/root/.checkers \

checkersd\_i:v1-alpine \

keys \

--keyring-backend file --keyring-dir /root/.checkers/keys \

show alice --address)

Copy

$ bob=$(echo password | docker run --rm -i \

-v $(pwd)/prod-sim/desk-bob:/root/.checkers \

checkersd\_i:v1-alpine \

keys \

--keyring-backend file --keyring-dir /root/.checkers/keys \

show bob --address)

The CosmJS tests use stake and token whereas this production setup uses only upawn. Therefore, do a text search and change all occurrences of stake and token to upawn in [client/test/integration/stored-game-action.ts (opens new window)↗](https://github.com/cosmos/academy-checkers-ui/blob/main/test/integration/stored-game-action.ts). Also remove the [upawn: 1, (opens new window)↗](https://github.com/cosmos/academy-checkers-ui/blob/main/test/integration/stored-game-action.ts#L56-L60) lines that prevent compilation.

Copying what you did in [leaderboard migration section](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/4-migration-leaderboard.html), credit the test accounts so that the CosmJS tests do not attempt to call a missing faucet:



Copy

$ echo password | docker run --rm -i \

-v $(pwd)/prod-sim/desk-alice:/root/.checkers \

--network checkers-prod\_net-public \

checkersd\_i:v1-alpine tx bank \

send $alice cosmos1fx6qlxwteeqxgxwsw83wkf4s9fcnnwk8z86sql 300upawn \

--from $alice \

--keyring-backend file --keyring-dir /root/.checkers/keys \

--chain-id checkers-1 \

--node http://node-carol:26657 \

--broadcast-mode block --yes

$ echo password | docker run --rm -i \

-v $(pwd)/prod-sim/desk-bob:/root/.checkers \

--network checkers-prod\_net-public \

checkersd\_i:v1-alpine tx bank \

send $bob cosmos1mql9aaux3453tdghk6rzkmk43stxvnvha4nv22 300upawn \

--from $bob \

--keyring-backend file --keyring-dir /root/.checkers/keys \

--chain-id checkers-1 \

--node http://node-carol:26657 \

--broadcast-mode block --yes

Then run the tests. One run of the tests creates one completed game. It is okay do this multiple times:



Copy

$ docker run --rm -it \

-v $(pwd)/client:/client \

-w /client \

--network checkers-prod\_net-public \

--env RPC\_URL="http://node-carol:26657" \

node:18.7-slim \

npm test

Note how the RPC\_URL is passed via an environment variable. This uses the fact that dotenv's config() function does not overwrite existing variables.

The completed game will count for the future leaderboard.

[#Copy link](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#prepare-the-upgrade-proposals) Prepare the upgrade proposals

Copying what was done in the [previous migration section](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/2-migration-info.html#governance-proposal), with one block every 5 seconds, you make:

* The first upgrade proposal to be run in 15 minutes (i.e. 180 blocks).
* The second upgrade proposal to be run in 25 minutes (i.e. 300 blocks).

Remember that both proposals will have a voting period of 10 minutes, with the second one straddling the first upgrade:

* At *t=0*, the first proposal is in its voting period, for an upgrade 15 minutes later (*t=+15 min*).
* At *t=+10 min*, the first proposal should pass, and at about the same time, you create the second proposal (it does not matter if it is a bit before or a bit after) for an upgrade 15 minutes later (*t=+25 min*).
* At *t=+15 min*, the first upgrade happens automatically thanks to Cosmovisor. You will now be running v1.1.
* At *t=+20 min*, the second proposal should pass.
* At *t=+25 min*, the second upgrade happens. You will now be running v2.

Find the current block height with:



Copy

$ docker run --rm -it \

--network checkers-prod\_net-public \

checkersd\_i:v1-alpine status \

--node "tcp://node-carol:26657" \

| jq -r ".SyncInfo.latest\_block\_height"

This returns something like:



Copy

1000

[#Copy link](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#send-the-first-upgrade-proposal) Send the first upgrade proposal

With a minimum deposit of 10,000,000 upawn, you can now have Alice send the governance proposal from her desktop to Carol's public node:



Copy

$ echo password | docker run --rm -i \

-v $(pwd)/prod-sim/desk-alice:/root/.checkers \

--network checkers-prod\_net-public \

checkersd\_i:v1-alpine \

tx gov submit-proposal software-upgrade v1tov1\_1 \

--node "tcp://node-carol:26657" \

--title "v1\_1tov2" \

--description "First step - add player infos" \

--upgrade-height 1180 \

--deposit 10000000upawn \

--from $alice --keyring-backend file --keyring-dir /root/.checkers/keys \

--broadcast-mode block \

--yes

The command is long but it makes sense when you look at it carefully. It returns you the proposal id:



Copy

- attributes:

- key: proposal\_id

value: "1"

- key: proposal\_type

value: SoftwareUpgrade

- key: voting\_period\_start

value: "1"

[#Copy link](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#vote-on-the-first-proposal) Vote on the first proposal

Have both Alice and Bob vote "yes" on the proposal:

**Alice**

**Bob**



Copy

echo password | docker run --rm -i \

-v $(pwd)/prod-sim/desk-alice:/root/.checkers \

--network checkers-prod\_net-public \

checkersd\_i:v1-alpine \

tx gov vote 1 yes \

--node "tcp://node-carol:26657" \

--from $alice --keyring-backend file --keyring-dir /root/.checkers/keys \

--yes

Copy

echo password | docker run --rm -i \

-v $(pwd)/prod-sim/desk-bob:/root/.checkers \

--network checkers-prod\_net-public \

checkersd\_i:v1-alpine \

tx gov vote 1 yes \

--node "tcp://node-carol:26657" \

--from $bob --keyring-backend file --keyring-dir /root/.checkers/keys \

--yes

[#Copy link](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#refill-your-cup) Refill your cup

Wait 10 minutes from the time you created the first proposal.

When the proposal voting period ends, check that the votes went through and what the latest block height is:

**Votes**

**Period**

**Block Height**



Copy

$ docker run --rm -it \

--network checkers-prod\_net-public \

checkersd\_i:v1-alpine \

query gov votes 1 \

--node "tcp://node-carol:26657"

Copy

$ docker run --rm -it \

--network checkers-prod\_net-public \

checkersd\_i:v1-alpine \

query gov proposal 1 \

--node "tcp://node-carol:26657"

Copy

$ docker run --rm -it \

--network checkers-prod\_net-public \

checkersd\_i:v1-alpine status \

--node "tcp://node-carol:26657" \

| jq -r ".SyncInfo.latest\_block\_height"

The proposal's current status will be:



Copy

status: PROPOSAL\_STATUS\_VOTING\_PERIOD

You must wait until after the proposal status changes to:



Copy

status: PROPOSAL\_STATUS\_PASSED

You must now wait longer, this time for the upgrade block to be reached. In the mean time...

[#Copy link](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#send-the-second-upgrade-proposal) Send the second upgrade proposal

Between the time the first proposal has passed and the first upgrade takes place, send the second proposal, this time by Bob:



Copy

$ echo password | docker run --rm -i \

-v $(pwd)/prod-sim/desk-bob:/root/.checkers \

--network checkers-prod\_net-public \

checkersd\_i:v1-alpine \

tx gov submit-proposal software-upgrade v1\_1tov2 \

--node "tcp://node-carol:26657" \

--title "v1\_1tov2" \

--description "Leaderboard introduction" \

--upgrade-height 1300 \

--deposit 10000000upawn \

--from $bob --keyring-backend file --keyring-dir /root/.checkers/keys \

--broadcast-mode block \

--yes

This returns you the second proposal id:



Copy

- attributes:

- key: proposal\_id

value: "2"

- key: proposal\_type

value: SoftwareUpgrade

- key: voting\_period\_start

value: "2"

Similarly, have Alice and Bob vote on it:

**Alice**

**Bob**



Copy

echo password | docker run --rm -i \

-v $(pwd)/prod-sim/desk-alice:/root/.checkers \

--network checkers-prod\_net-public \

checkersd\_i:v1-alpine \

tx gov vote 2 yes \

--node "tcp://node-carol:26657" \

--from $alice --keyring-backend file --keyring-dir /root/.checkers/keys \

--yes

Copy

echo password | docker run --rm -i \

-v $(pwd)/prod-sim/desk-bob:/root/.checkers \

--network checkers-prod\_net-public \

checkersd\_i:v1-alpine \

tx gov vote 2 yes \

--node "tcp://node-carol:26657" \

--from $bob --keyring-backend file --keyring-dir /root/.checkers/keys \

--yes

[#Copy link](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#the-first-live-upgrade) The first live upgrade

If you are scanning the logs of one of the containers, for instance from Docker's GUI, you should see something like:



Copy

ERR UPGRADE "v1tov1\_1" NEEDED at height: 1180:

INF starting node with ABCI Tendermint in-process

That was v1's last message, followed by v1\_1's first message.

After that, you should be able to query for the presence of player infos:



Copy

$ docker run --rm -it \

--network checkers-prod\_net-public \

checkersd\_i:v1.1-alpine \

query checkers list-player-info \

--node "tcp://node-carol:26657"

This should return:



Copy

playerInfo:

- forfeitedCount: "0"

index: cosmos1fx6qlxwteeqxgxwsw83wkf4s9fcnnwk8z86sql

lostCount: "0"

wonCount: "2"

- forfeitedCount: "0"

index: cosmos1mql9aaux3453tdghk6rzkmk43stxvnvha4nv22

lostCount: "2"

You can also confirm that the leaderboard is still missing:



Copy

$ docker run --rm -it \

--network checkers-prod\_net-public \

checkersd\_i:v2-alpine \

query leaderboard show-leaderboard \

--node "tcp://node-carol:26657"

This still returns:



Copy

Error: rpc error: code = Unknown desc = unknown query path: unknown request

You can also verify with Cosmosvisor that it is now running v1.1:



Copy

$ docker exec -it node-carol \

ls -l /root/.checkers-upgrade/cosmovisor/current

This should return:



Copy

... /root/.checkers-upgrade/cosmovisor/current -> /root/.checkers-upgrade/cosmovisor/upgrade/v1tov1\_1

[#Copy link](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#the-second-live-upgrade) The second live upgrade

You can follow the status of the second proposal with:



Copy

$ docker run --rm -it \

--network checkers-prod\_net-public \

checkersd\_i:v1.1-alpine \

query gov proposal 2 \

--node "tcp://node-carol:26657"

Observe that it changes to PASSED after the first upgrade. After that, if you are scanning the logs of one of the containers (for instance from Docker's GUI) you should see something like:



Copy

ERR UPGRADE "v1\_1tov2" NEEDED at height: 1300:

INF starting node with ABCI Tendermint in-process

That was v1.1's last message, followed by v2's first message. You can confirm that the leaderboard has been populated:



Copy

$ docker run --rm -it \

--network checkers-prod\_net-public \

checkersd\_i:v2-alpine \

query leaderboard show-leaderboard \

--node "tcp://node-carol:26657"

This returns:



Copy

Leaderboard:

winners:

- addedAt: "1683213831"

address: cosmos1fx6qlxwteeqxgxwsw83wkf4s9fcnnwk8z86sql

wonCount: "2"

[#Copy link](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#what-about-stop-and-restart) What about stop and restart?

All checkers containers are now running checkersd v2. You can see that Cosmovisor has swapped the current executable:



Copy

$ docker exec -it node-carol \

ls -l /root/.checkers-upgrade/cosmovisor/current

This should return:



Copy

... /root/.checkers-upgrade/cosmovisor/current -> /root/.checkers-upgrade/cosmovisor/upgrade/v1\_1tov2

It will return this **until the containers are stopped** and deleted, that is.



Remember that the containers are loaded from a Docker image configured with Cosmovisor. In the current configuration, Cosmovisor starts with what it finds at genesis/bin/checkersd, i.e. v1.   
  
All this is to say that you should not expect it to work if you stop and start your Cosmovisor Compose setup as is.

If you were using real production servers, Cosmovisor's symbolic link would not reset itself on restart, so you would be safe in this regard. You would have time to revisit your server's configuration so as to launch checkersd v2 natively.

In this example you can prepare yet another Compose file, this time specifically for v2:



Copy

version: "3.7"

services:

val-alice:

extends:

file: docker-compose.yml

service: val-alice

image: checkersd\_i:v2-alpine

sentry-alice:

extends:

file: docker-compose.yml

service: sentry-alice

image: checkersd\_i:v2-alpine

val-bob:

extends:

file: docker-compose.yml

service: val-bob

image: checkersd\_i:v2-alpine

sentry-bob:

extends:

file: docker-compose.yml

service: sentry-bob

image: checkersd\_i:v2-alpine

node-carol:

extends:

file: docker-compose.yml

service: node-carol

image: checkersd\_i:v2-alpine

prod-sim /

docker-compose-v2.yml

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/migration-prod/prod-sim/docker-compose-v2.yml" \t "_blank)

Now you can safely stop the Cosmovisor setup:



Copy

$ docker compose --project-name checkers-prod down

And start the v2 setup:



Copy

$ docker compose \

--file prod-sim/docker-compose.yml \

--file prod-sim/docker-compose-v2.yml \

--project-name checkers-prod up \

--detach



If you want to test other migration configurations, for instance where Carol *forgot* to put Cosmovisor on her node, you can revert all your blockchain files to v1 with:



Copy

$ ./prod-sim/unsafe-reset-state.sh

Another exercise you can attempt is create a v1tov2 upgrade that does **both** upgrades in one go. You would have to add this v1tov2 name into the Go code, and make sure it is handled correctly.

synopsis

To summarize, this section has explored:

* How to prepare multi-stage Docker images for different executable versions.
* How to prepare Cosmovisor for a simulated production migration.
* How to upgrade a blockchain in production, by live migrating from v1 of the blockchain to v1.1 and then v2.
* How to launch all that with the help of Docker Compose.
* A complete procedure for how to conduct the update via the CLI.

previous

[](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/4-migration-leaderboard.html)

**[Migrate the Leaderboard Module After Production](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/4-migration-leaderboard.html)**

up next

**[Final Exam](https://ida.interchain.io/ida-course/final-exam/)**

[[](https://ida.interchain.io/ida-course/final-exam/)](https://ida.interchain.io/ida-course/final-exam/)

Rate this Page

icon smile

icon meh

icon frown

Would you like to add a message?

Submit

Thank you for your Feedback!

[](https://ida.interchain.io/ida-course/discord-info.html)

On this page

[What to expect](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#what-to-expect)

[Prepare checkers executables](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#prepare-checkers-executables)

[Blockchain elements](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#blockchain-elements)

[Prepare the Cosmovisor executable](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#prepare-the-cosmovisor-executable)

[Docker Compose elements](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#docker-compose-elements)

[Run all the elements](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#run-all-the-elements)

[Add games](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#add-games)

[Prepare the upgrade proposals](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#prepare-the-upgrade-proposals)

[Send the first upgrade proposal](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#send-the-first-upgrade-proposal)

[Vote on the first proposal](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#vote-on-the-first-proposal)

[Refill your cup](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#refill-your-cup)

[Send the second upgrade proposal](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#send-the-second-upgrade-proposal)

[The first live upgrade](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#the-first-live-upgrade)

[The second live upgrade](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#the-second-live-upgrade)

[What about stop and restart?](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/5-migration-prod.html#what-about-stop-and-restart)

#### **Get Cosmos updates**

Unsubscribe at any time. [Privacy Policy↗](https://v1.cosmos.network/privacy)

     Next

Documentation

[Cosmos SDK](https://docs.cosmos.network/)[Cosmos Hub](https://hub.cosmos.network/)[CometBFT](https://docs.cometbft.com/)[IBC Protocol](https://ibc.cosmos.network/)

Community

[Interchain blog](https://blog.cosmos.network/)[Forum](https://forum.cosmos.network/)[Discord](https://discord.gg/cosmosnetwork)

Contributing

[Source code on GitHub](https://github.com/cosmos/sdk-tutorials)

[](https://ida.interchain.io/)

[Interchain Developer Academy](https://ida.interchain.io/)

**[](https://blog.cosmos.network/)[](https://twitter.com/cosmos)[](https://discord.gg/cosmosnetwork)[](https://www.linkedin.com/company/interchain-foundation/about/)[](https://reddit.com/r/cosmosnetwork)[](https://t.me/cosmosproject)[](https://www.youtube.com/c/CosmosProject)**



Dark mode

† This website is maintained by the Interchain Foundation (ICF). The contents and opinions of this website are those of the ICF. The ICF provides links to cryptocurrency exchanges as a service to the public. The ICF does not warrant that the information provided by these websites is correct, complete, and up-to-date. The ICF is not responsible for their content and expressly rejects any liability for damages of any kind resulting from the use, reference to, or reliance on any information contained within these websites.

Cosmos is a registered trademark of the [Interchain Foundation.](https://interchain.io/)[Privacy](https://v1.cosmos.network/privacy)