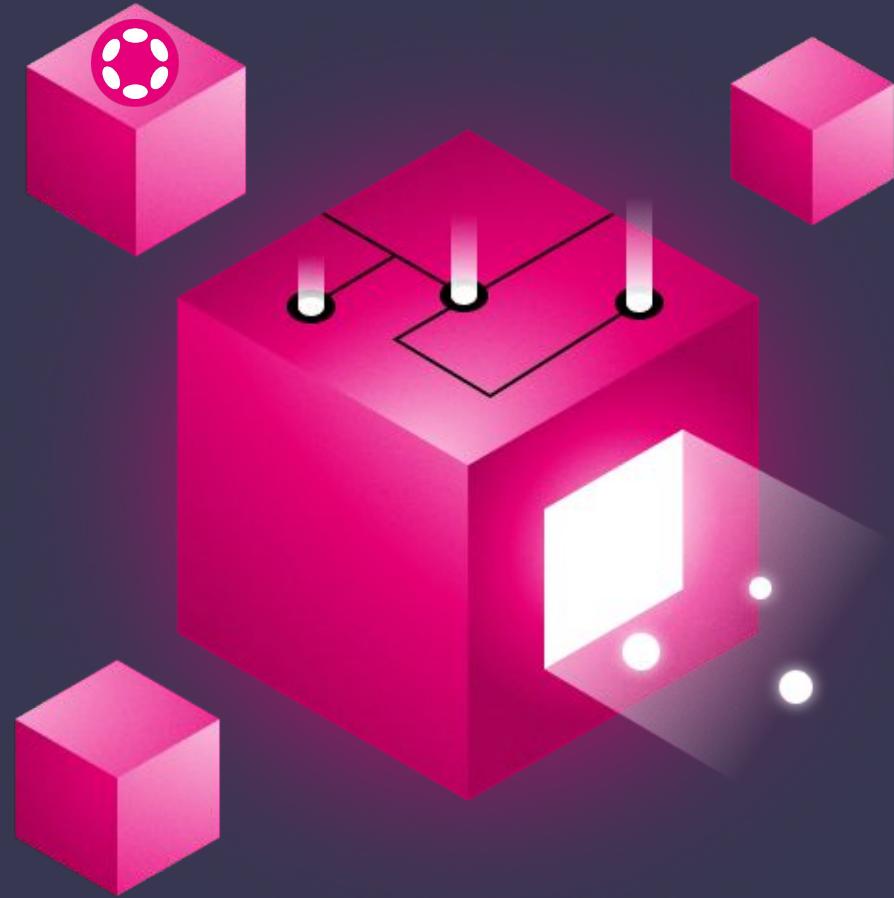




DEPLOYING POLKADOT AND PARACHAIN TESTNETS ON KUBERNETES



Pierre Besson
@pibesson
@pierre.besson@matrix.parity.io

Pierre Besson

- DevOps Engineer at Parity Technologies
- Blockchain node operator for Parity's testnets
- Initiator of the Parity helm-chart and Testnet-manager open source projects



Contact

⟳ Element: @pierre.besson:matrix.parity.io

🐦 Twitter: @pibesson



WHAT THIS PRESENTATION IS ABOUT ?

01

TESTNET OPERATIONS AT PARITY

Tell the story of
operating
increasingly bigger
testnets

02

PARITY KUBERNETES TESTNET STACK

Present the
Kubernetes-based open
source stack that we use
to operate our testnets

03

WHAT WE LEARNED

Our personal feedback
and vision of blockchain
node operations

testnet Operations at Parity



WHY DO we need TESTNETS ?

Testnet: simulation of a “value bearing” network for validation and debugging

For chain users :

- Sending transactions for free !
- Testing interoperability

For chain developers :

Relay (or solo) chains :

- Testing node client upgrades
- Validating runtime upgrades
- Comprehensively observe the network

Parachains :

- Dry-run their parachain onboarding
- Have access to relay-chain logs relevant to their runtime execution

DEVOPS IN THE CONTEXT OF BLOCKCHAIN OPERATIONS

Blockchain nodes have unique operational concerns:

-  Stateful databases at core
-  Distributed systems, p2p networking
-  High security requirements for production nodes
-  Key management
-  Monitoring

Challenges of Testnets:

- Quickly scale up/down the number of nodes
- Upgrade more often
- Runs untested code

 Breaking consensus → hardfork or full reset of the network

BEGINNINGS OF TESTNETS AT PARITY

 Configuration management (Ansible) used for managing nodes

Ansible collection (open source): github.com/paritytech/ansible-galaxy

Pros:

- Convenient as configuration is the same for dev/prod
- Secure as we connect only from personal laptop (no CI)

Cons:

- Slow for managing a lot of nodes
- No developer self-service
- Low automation

GROWING TESTNET REQUIREMENTS

Need to scale:

- Number of developers to support
- Number of networks (westend + rococo + dozens of parachains)
- Temporary scaling to 1000 validators
- Lowering time to answers deployment requests

Solution:

- Deploying testnet nodes in Kubernetes
- Creation of our helm-chart collection (open source)
github.com/paritytech/helm-charts
- Scripting node keys management
github.com/paritytech/testnet-manager

Parity Kubernetes Testnet Stack

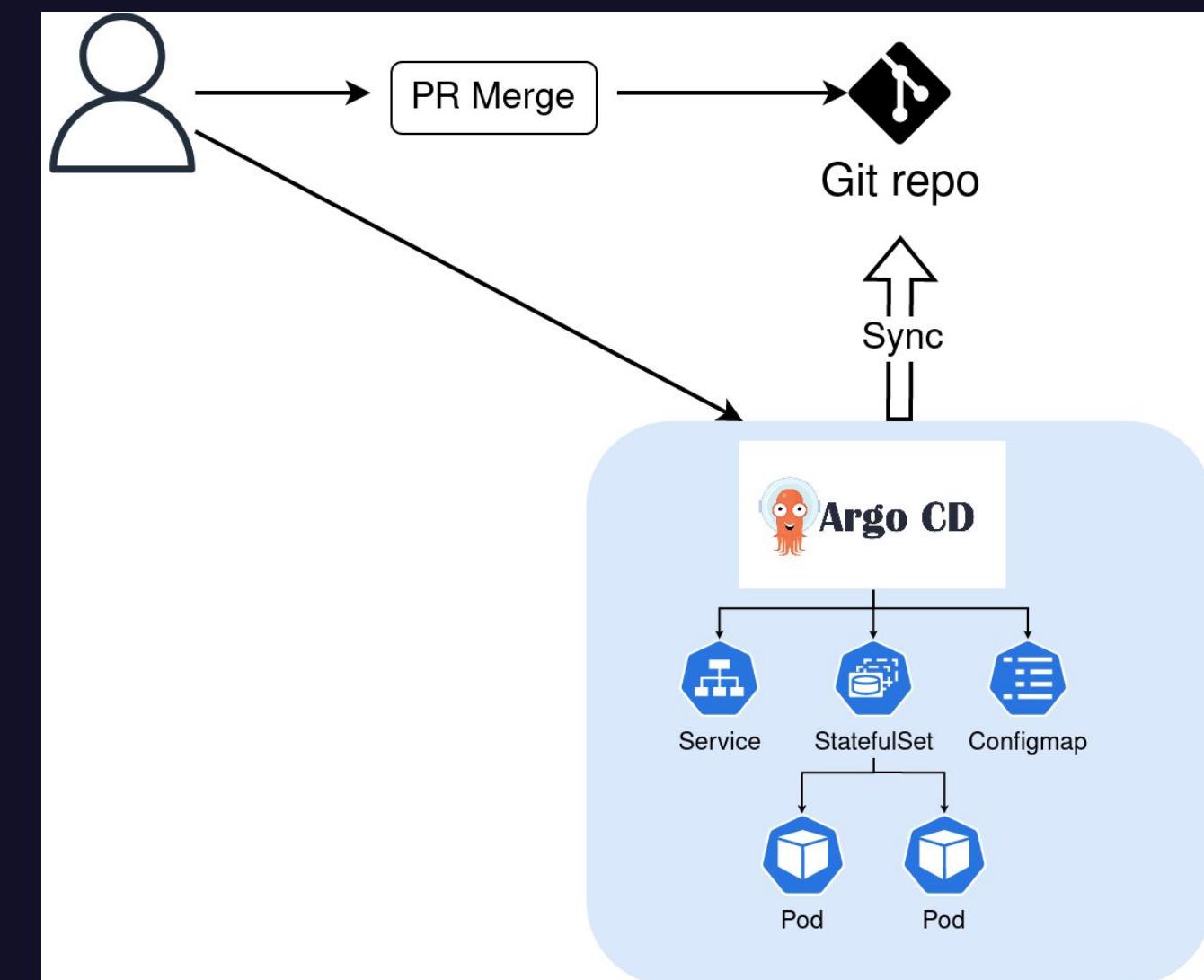


WHY KUBERNETES ?

- Most mature container orchestration platform
- Familiar to DevOps practitioners
- Support for advanced networking and stateful workloads
- Persistent volume management
- Declarative approach:
 - Automate operations (restart, rolling upgrades, ...)
 - Self-healing (healthcheck, rollbacks on startup failure)
 - Reuse configuration across networks
- Abstract config complexity with Helm
- Integrate with our monitoring stack (Prometheus operator, Loki)

WHY GITOPS ?

- Git repo as the source of truth of the system **desired state**
- Different from using a CI pipeline to push infrastructure changes
- In-cluster operator **reconcile** the actual and desired states





paritytech / **helm-charts** Public

Edit Pins ▾ Unwatch 10 ▾ Fork 7 ▾ Starred 18 ▾

Code Issues 13 Pull requests 2 Actions Projects Wiki Security Insights Settings

README.md

Parity Helm Charts

Parity's [Kubernetes Helm](#) charts collection.

Charts list

- [Common](#): a generic helm chart for kubernetes
- [Node](#): deploy Substrate/Polkadot nodes
- [Parity Bridge Common](#): deploy parity-bridge-common service
- [Polkadot Basic Notification](#): deploy a chain notification bot
- [Polkadot Introspector](#): deploy a chain monitoring and introspection service
- [Polkadot Runtime Exporter](#): deploy a tool to collect runtime statistics
- [Staking miner](#): deploy the staking-miner for submitting solutions to NPoS elections
- [Substrate faucet](#): deploy Substrate Faucet service
- [Substrate telemetry](#): deploy the Substrate Telemetry service
- [Testnet Manager](#): deploy a management tool for operating testnets

About

Parity Helm charts collection

Readme
GPL-3.0 license
18 stars
10 watching
7 forks

Releases 151

[node-4.2.4](#) Latest
6 days ago
+ 150 releases

Packages

No packages published
[Publish your first package](#)

Contributors 10

THE “node” HELM-CHART

Deploy any kind of substrate node (full, validator, collator)

- Manage nodes using StatefulSets (ordered set of pods with same config)
- Support Relay-chain and Para-chain (collator) nodes
- Expose p2p ports through a Kubernetes Service
- Inject keys into node from a Secret or Hashicorp Vault
- Expose RPC endpoint through an Ingress
- Retrieve chain data from a snapshot (HTTP URL, GCS bucket, VolumeSnapshot)



EXAMPLE: 10 ROCOCO VALIDATORS

```
image:
  repository: parity/polkadot
  tag: v0.9.31

node:
  chain: rococo
  role: authority
  replicas: 10
  chainData:
    database: paritydb
    pruning: 1000
    volumeSize: 100Gi
  flags:
    - "--beefy"
  logLevels:
    - "parachain=debug"
resources:
  requests:
    cpu: "3800m"
    memory: "4Gi"
  limits:
    cpu: "7500m"
    memory: "8Gi"
```



EXAMPLE: 4 ROCKMINE COLLATORS

```
● ○ ●

image:
  repository: parity/polkadot-parachain
  tag: 0.9.320

node:
  chain: rococo-rockmine
  role: collator
  replicas: 4
  command: polkadot-parachain
  customChainspecUrl: "https://paritytech.github.io
/chainspecs/rococo/parachain/rockmine/chainspec.json"
  chainData:
    pruning: 1000
    database: paritydb
    volumeSize: 50Gi
  chainKeystore:
    mountInMemory:
      enabled: true
  isParachain: true
  collatorRelayChain:
    chain: rococo
    customChainspecUrl: "https://paritytech.github.io
/chainspecs/rococo/relaychain/chainspec.json"
    chainData:
      pruning: 1000
      database: paritydb
      volumeSize: 100Gi
      GCSBucketUrl: "gs://my-backup-bucket"
```



key injection + Load-Balancer

```
● ● ●

node:
  chain: rococo-custom
  role: authority
  replicas: 4
  [...]
  keys:
    - type: aura
      scheme: sr25519
      seed: "test test test test test test test test test test junk"
      # create derived key for each pod
      extraDerivation: "//${HOSTNAME}//aura"
    - type: babe
      scheme: ed25519
      seed: "test test test test test test test test test test junk"
      extraDerivation: "//${HOSTNAME}//aura"
    [...]
  perNodeServices:
    relayP2pService:
      enabled: true
      type: LoadBalancer
    externalDns:
      enabled: true
      hostname: rococo.example.com.
```

TOOLING HELM-CHARTS

Additional tooling helm-charts are also part of the collection:

- Faucet
- Notification-bot
- Staking-miner
- Telemetry
- ...

(SEMI-)AUTOMATING NODE OPERATIONS

Just deploying nodes is not enough, there are also operations to perform:

eg. add extra validators/collators to a network:

- On the node: generate keys (rotate_key)
- On chain: submit setKeys and Sudo extrinsics

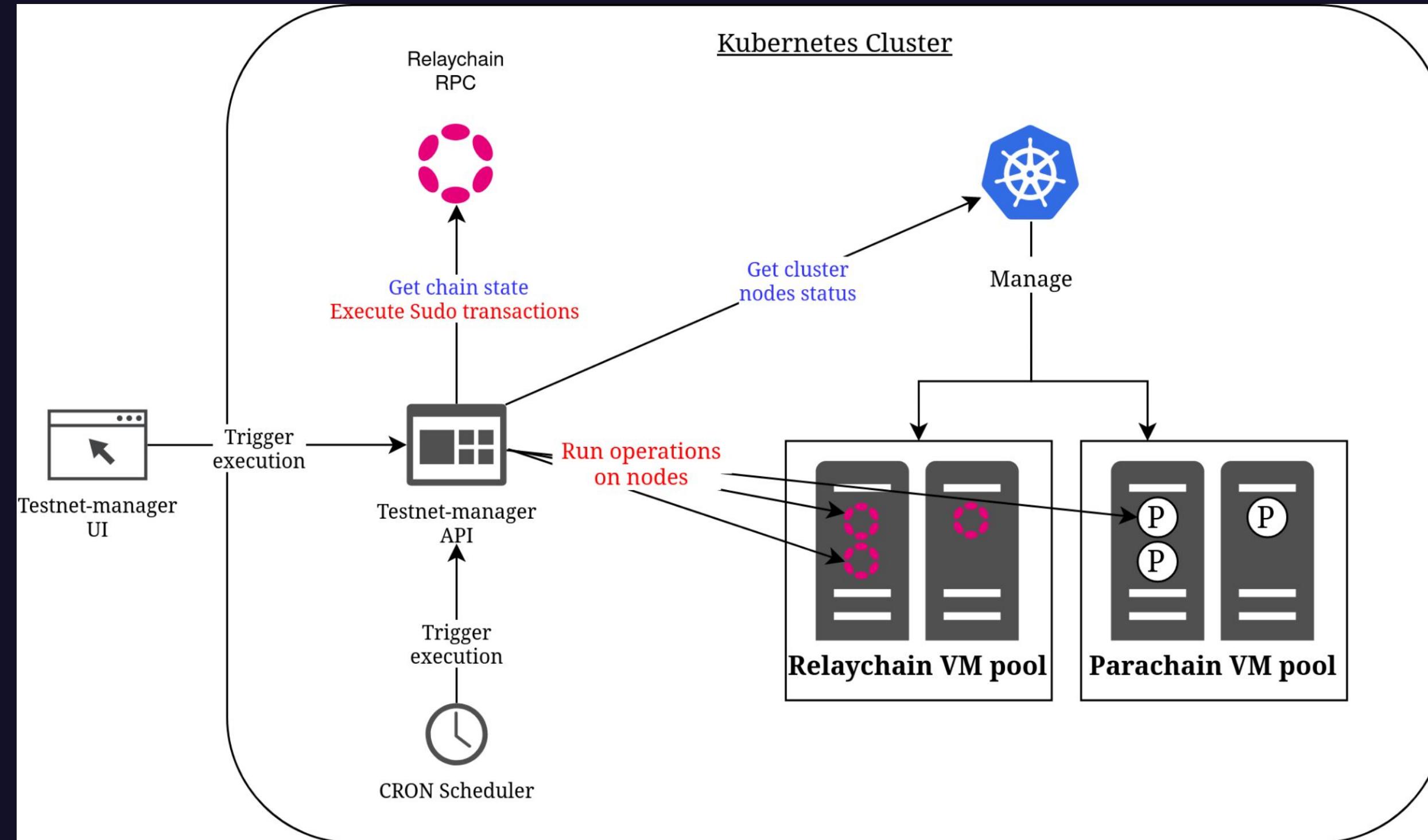
Additional issues:

- Keeping track of active/inactive validators in a dynamic environment
- Facilitating onboarding/offboarding of parachains

TESTNET-manager FEATURES

- List validators and collators in the namespace
- Map nodes to their on-chain address
- List active parachains from on-chain state
- Detect namespace collators corresponding to a parachain
- Check node readiness
- Orchestrate and parallelize node operations (eg. session keys)
- Register/deregister validators and collators in batch

testnet-manager architecture



testnet-manager

0.1.0

OAS3

/openapi.json

default

GET /api/nodes Get Nodes**GET** /api/nodes/{node_name} Get Nodes**GET** /api/validators Get Validators**GET** /api/parachains Get Parachains**GET** /api/collators/{para_id} Get Collators**POST** /api/register_validators Register Validators**POST** /api/deregister_validators Deregister Validators**POST** /api/rotate_session_keys Rotate Session Keys**POST** /api/onboard_parachain/{para_id} Onboard Parachain**POST** /api/offboard_parachain/{para_id} Offboard Parachain**POST** /api/register_collators/{para_id} Register Collators**POST** /api/deregister_collators/{para_id} Deregister Collators**GET** /health Health

VALIDATORS VIEW

Rococo Validators : 0 Active in VM / 101 Active in K8S / 1 Inactive in K8S

Logs		Address	Subscan	Version	Status	Location	Validator	Keys
Name	Logs							
rococo-validator-a-node-0		5EXam2LBfS8werJtepdoWhjdfL7MygV7y5Kav1NxYmupNewT		parity/polkadot:v0.9.31	Running	in_cluster	True	
rococo-validator-a-node-1		5CUMBddkTEkFC7GoBGmfMhXo7UXicudZSQDBByfv4y9yGEa5j		parity/polkadot:v0.9.31	Running	in_cluster	True	
rococo-validator-a-node-2		5ETmQC23TZuKrgZVD7rCPqAvVZhSB4SwRBqqHkDQaABBHueT		parity/polkadot:v0.9.31	Running	in_cluster	True	
rococo-validator-a-node-3		5HVp6vPrhGtvX1KNEaEgN9Wf7QhsnXFcPAQGurbsgsd5L8dK		parity/polkadot:v0.9.31	Running	in_cluster	True	
rococo-validator-a-node-4		5EbzGeJrc4ui8YewRwSVYtUmpqwZjYN6Y2DQseztBPqnKkbR		parity/polkadot:v0.9.31	Running	in_cluster	True	
rococo-validator-a-node-5		5EkES12da7vGsaz1ZZ7BZ9UeJJLuAfiosWJQutfP1fBNM6qp		parity/polkadot:v0.9.31	Running	in_cluster	True	
rococo-validator-a-node-6		5CJXvichGEEwtHioB9tabX7EKgNd9J5cz2LHK6dZw2CrqMqb		parity/polkadot:v0.9.31	Running	in_cluster	True	
rococo-validator-a-node-7		5EF1JnrXWWWDQrQCHhL77Hxy3WwxVzDv8JBwj6oDeZTd3FgLc		parity/polkadot:v0.9.31	Running	in_cluster	True	
rococo-validator-a-node-8		5CFz1ZJYTJydqPVJQMofoDMWFZfVkCRz2bHDzMNuiA7y86GY		parity/polkadot:v0.9.31	Running	in_cluster	True	
rococo-validator-a-node-9		5EYzUHNiwMQ3AUX2h1r2FYqtUZHk8nTpkYWSuv19WBFLfaV		parity/polkadot:v0.9.31	Running	in_cluster	True	

Showing 1 to 10 of 10 entries (filtered from 102 total entries)

Previous 1 Next



subØ

VALIDATORS VIEW

Rococo Validators : 0 Active in VM / 101 Active in K8S / 1 Inactive in K8S

all		Rotate session keys						Search: validator-a	Keys
Name	Logs	Address	Subscan	Version	Status	Location	Validator		
rococo-validator-a-node-0	🔗	5EXam2LBfS8werJtepdoWhjdfL7MygV7y5Kav1NxYmupNewT	🔗	parity/polkadot:v0.9.32	Running	in_cluster	True	🔑	
rococo-validator-a-node-1	🔗	5CUMBddkTEkFC7GoBGmfMhXo7UXicudZSQDByfv4y9yGEa5j	🔗	parity/polkadot:v0.9.32	Running	in_cluster	True	🔑	
rococo-validator-a-node-2	🔗	5ETmQC23TZuKrgZVD7rCPqAvVZhSB4SwRBqqHkDQaABBHueT	🔗	parity/polkadot:v0.9.32	Running	in_cluster	True	🔑	
rococo-validator-a-node-3	🔗	5HVp6vPrhGtvX1KNEaEgN9Wf7QhsnXFcPAQGurbsgsd5L8dK	🔗	parity/polkadot:v0.9.32	Running	in_cluster	True	 <ul style="list-style-type: none">grandpa: 0xbc3da530f4c42721104b7d6be84611a6edfa461484bc2ebabe: 0xb4564f195f762cf168764c4be118d463a60b494c94801fcim_online: 0x16bd6fa867e24627ddf538c38e53e6778364f2354780ef2para_validator: 0x0e663b465270824348e57fb87a882fc0a744bd72a413aepara_assignment: 0x46f2b962aa89e0fc48fc08a7fc7b52e9ea95f12ec09b589authority_discovery: 0xe8ae81c4639f2a50e9eac251484c87124dcbe18deee342beefy: 0x0350f9db83a51c2ab5411e53deb9071454ceeeec3ef736	
rococo-validator-a-node-4	🔗	5EbzGeJrc4ui8YewRwSVYtUmpqwZjYN6Y2DQseztBPqnKkbR	🔗	parity/polkadot:v0.9.32	Running	in_cluster	True	🔑	

VALIDATORS STATEFUL SET VIEW

Rococo Validators : 0 Active in VM / 10 Active in K8S						
rococo-validator-a-node	Register	Deregister	Rotate session keys	Search:		
	Address	Subscan	Version	Status	Location	Validator
all						
rococo-adrian-node						
rococo-validator-a-node						
rococo-validator-b-node						
rococo-validator-c-node						
rococo-validator-d-node						
rococo-validator-e-node						
rococo-validator-f-node						
rococo-validator-g-node						
rococo-validator-h-node						
rococo-validator-i-node						
rococo-validator-j-node						
rococo-validator-profile-node						
rococo-validator-a-node-1	5CUMBddkTEkFC7GoBGmfMhXo7UXicudZSQDByfv4y9yGEa5j	🔗	parity/polkadot:v0.9.32	Running	in_cluster	True
rococo-validator-a-node-2	5ETmQC23TZuKrgZVD7rCPqAvVZhSB4SwRBqqHkDQaABBHueT	🔗	parity/polkadot:v0.9.32	Running	in_cluster	True
rococo-validator-a-node-3	5HVp6vPrhGtvX1KNEaEgN9Wf7QhsnXFcPAQGurbsgsd5L8dK	🔗	parity/polkadot:v0.9.32	Running	in_cluster	True
rococo-validator-a-node-4	5EbzGeJrc4ui8YewRwSVYtUmpqwZjYN6Y2DQseztBPqnKkbR	🔗	parity/polkadot:v0.9.32	Running	in_cluster	True



SINGLE VALIDATOR VIEW

sub0

rococo-validator-a-node-0	
Deregister Validator Rotate session keys	
Show All entries	
Keys	Values
Args	
Chain	rococo
Image	parity/polkadot:v0.9.31
Is Syncing	False
Labels	
Local Listen Addresses	
Peer ID	12D3KooWQGxHHbV4FCzPzE1zK2WmsWy8H8bmLDk253iZCWqAiYmJ
Peers	
Peers Count	40
Properties	{'ss58Format': 42, 'tokenDecimals': 12, 'tokenSymbol': 'ROC'}
Ready	True
Roles	Authority
Substrate Status	Running
Sync State	{'startingBlock': 2701013, 'currentBlock': 2732664, 'highestBlock': 2732664}
Uptime	2 days, 4:49:13
Validator Account	5EXam2LBfS8werJtepdoWhjdfL7MygV7y5Kav1NxYmupNewT
Validator Active	True
Version	0.9.31-32dd0c9cfcd
Showing 1 to 18 of 18 entries	
Previous 1 Next	



subØ

Parachains view

Rococo : 137 Paras: 42 Parachains, 95 Parathreads, 0 Others

Show 30 entries

Search:

ParaID	Name	Lifecycle	Location	Leases	Actions	Head	CurrentCodeHash
1000	rococo-rockmine 🌏	Parachain	in_cluster	824	Offboard	★	#
1002	rococo-para-1002	Parachain	external	176	Offboard	★	#
1003	rococo-para-1003	Parachain	external	176	Offboard	★	#
1013	rococo-para-1013	Parachain	external	820	Offboard	★	#
1900	rococo-dummy 🌐	Parachain	in_cluster	303	Offboard	★	#
2000	rococo-para-2000	Parachain	external	811	Offboard	★	#
2004	rococo-para-2004	Parachain	external	176	Offboard	★	#
2006	rococo-para-2006	Parachain	external	194	Offboard	★	#
2007	rococo-para-2007	Parachain	external	7	Offboard	★	#
2011	rococo-para-2011	Parathread	external	None	Offboard	★	#
2012	rococo-para-2012	Parachain	external	260	Offboard	★	#
2015	rococo-para-2015	Parachain	external	177	Offboard	★	#
2016	rococo-para-2016	Parachain	external	294	Offboard	★	#
2021	rococo-para-2021	Parachain	external	264	Offboard	★	#
2024	rococo-para-2024	Parachain	external	176	Offboard	★	#
2026	rococo-para-2026	Parachain	external	176	Offboard	★	#
2030	rococo-para-2030	Parachain	external	338	Offboard	★	#
2031	rococo-para-2031	Parachain	external	176	Offboard	★	#



COLLATORS VIEW

subØ

Rococo-rockmine [Rococo Para #1000] Collators : 4 External / 1 Active in K8S / 1 Inactive in K8S

Runtime: 9290, Status: Parachain, Desired number of candidates: 1

Name	Logs	Account	PodStatus	CollatorStatus	Image	Location
rococo-rockmine-collator-node-0		E8XC6rTJRsiOKCp6KMy6zd24ykj4gWsusZ3AkSeyavpVBAG	Running	False	parity/polkadot-parachain:0.9.290	in_cluster
rococo-rockmine-collator-only-rpc-node-0		EDCs28tXX3MyZLscDpXLWEwxtMe5vj9GckTeQwsVbSVZkwA	Running	Candidate	paritypr/polkadot-parachain-debug:master-7612d616	in_cluster
unknown-rococo-rockmine-0		EahDPH8dNNsSW8FaN5cbjPJ4GLd8GL4S84b89nxRYmG9BWf	?	Invulnerable	?	external
unknown-rococo-rockmine-1		GV6iK3WWf76JKMsmkSrJSE7fwZKDoRCmL27ZyCCfP5bCT8h	?	Invulnerable	?	external
unknown-rococo-rockmine-2		HpXiyogSCQpwFQA49oCmyiSJ DwCkzbkroHQ5ygWHCuCxRUN	?	Invulnerable	?	external
unknown-rococo-rockmine-3		JHsf5YvwXqatWnMjYqWNuWNqgNrEMZJAqp25P6LQ5SAmtRM	?	Invulnerable	?	external

Showing 1 to 6 of 6 entries

Previous 1 Next

SINGLE COLLATOR VIEW

rococo-rockmine-collator-node-0	
Register Collator	
Show	All ▾ entries
Keys	▲
Keys	Values
Args	⚙️
Chain	rococo-rockmine
Collator Account	E8XC6rTJRsiOKCp6KMy6zd24ykj4gWsusZ3AkSeyavpVBAG
Collator Status	False
Image	parity/polkadot-parachain:0.9.290
Is Syncing	False
Labels	🔗
Local Listen Addresses	💻
Para ID	1000
Peer ID	12D3KooWRrZMndHAopzao34uGsN7srjS3gh9nAjTGKLSyJeU31Lg
Peers	🤝
Peers Count	3
Properties	{'tokenDecimals': 12, 'tokenSymbol': 'ROC'}
Ready	True
Roles	Authority
Substrate Status	Running
Sync State	{'startingBlock': 1011582, 'currentBlock': 1025521, 'highestBlock': 1025521}
Uptime	1 day, 23:05:45
Version	0.9.290-4271ac75d34
Showing 1 to 19 of 19 entries	
Previous 1 Next	

what we
learned ?



PERSONAL FEEDBACK

- Operating each node “individually” is not a good approach for testnets
- Kubernetes can be the right platform even for stateful workloads
 - Increase deployment speed
 - Facilitate experimentation
- But it has significant drawbacks:
 - Introduces a lot of complexity
 - Reduces security and ease of troubleshooting
 - Create a differentiation between staging/production

Conclusion: use the right tool for the job

ADAPTING YOUR BLOCKCHAIN INFRASTRUCTURE TO REQUIREMENTS

- For local deployments, Kubernetes is overkill -> use Zombienet
- Choose the tool that provide the right tradeoffs
- Ask the right questions:
 - **Temporary or Persistent network ?**
 - **Network size** ? Will it grow over time ?
 - **Security vs Flexibility** ?
 - Developer self-service ?
- Hard problems:
 - Monitoring / Observability
 - Decentralization vs Control
 - Minimize downtime when deploying new code

TESTNET OPERATIONAL MODELS

Environment	Local/CI	Staging	Production
Network Type	"Temp" Network	Testnet	Mainnet
Automation Level	Fully Automated	Semi Automated	Manual
Operational Model	Throwaway	Cattle	Pet
Control on infrastructure	No control on infra specs	Fine tune infra specs	Strictly follow requirements
Recommended Deployment tool	Zombienet	Kubernetes + Testnet-manager	Ansible
Number of nodes	Few dozens	Hundreds	As much as possible

TESTNET OPERATIONAL MODELS

Environment	Local/CI	Staging	Production
Network Type	"Temp" Network	Testnet	Mainnet
Automation Level	Fully Automated	Semi Automated	Manual
Operational Model	Throwaway	Cattle	Pet
Control on infrastructure	No control on infra specs	Fine tune infra specs	Strictly follow requirements
Recommended Deployment tool	Zombienet	Kubernetes + Testnet-manager	Ansible
Number of nodes	Few dozens	Hundreds	As much as possible

TESTNET OPERATIONAL MODELS

Environment	Local/CI	Staging	Production
Network Type	"Temp" Network	Testnet	Mainnet
Automation Level	Fully Automated	Semi Automated	Manual
Operational Model	Throwaway	Cattle	Pet
Control on infrastructure	No control on infra specs	Fine tune infra specs	Strictly follow requirements
Recommended Deployment tool	Zombienet	Kubernetes + Testnet-manager	Ansible
Number of nodes	Few dozens	Hundreds	As much as possible

TESTNET OPERATIONAL MODELS

Environment	Local/CI	Staging	Production
Network Type	"Temp" Network	Testnet	Mainnet
Automation Level	Fully Automated	Semi Automated	Manual
Operational Model	Throwaway	Cattle	Pet
Control on infrastructure	No control on infra specs	Fine tune infra specs	Strictly follow requirements
Recommended Deployment tool	Zombienet	Kubernetes + Testnet-manager	Ansible
Number of nodes	Few dozens	Hundreds	As much as possible

TESTNET OPERATIONAL MODELS

Environment	Local/CI	Staging	Production
Network Type	"Temp" Network	Testnet	Mainnet
Automation Level	Fully Automated	Semi Automated	Manual
Operational Model	Throwaway	Cattle	Pet
Control on infrastructure	No control on infra specs	Fine tune infra specs	Strictly follow requirements
Recommended Deployment tool	Zombienet	Kubernetes + Testnet-manager	Ansible
Number of nodes	Few dozens	Hundreds	As much as possible

TESTNET OPERATIONAL MODELS

Environment	Local/CI	Staging	Production
Network Type	"Temp" Network	Testnet	Mainnet
Automation Level	Fully Automated	Semi Automated	Manual
Operational Model	Throwaway	Cattle	Pet
Control on infrastructure	No control on infra specs	Fine tune infra specs	Strictly follow requirements
Recommended Deployment tool	Zombienet	Kubernetes + Testnet-manager	Ansible
Number of nodes	Few dozens	Hundreds	As much as possible

TESTNET OPERATIONAL MODELS

Environment	Local/CI	Staging	Production
Network Type	"Temp" Network	Testnet	Mainnet
Automation Level	Fully Automated	Semi Automated	Manual
Operational Model	Throwaway	Cattle	Pet
Control on infrastructure	No control on infra specs	Fine tune infra specs	Strictly follow requirements
Recommended Deployment tool	Zombienet	Kubernetes + Testnet-manager	Ansible
Number of nodes	Few dozens	Hundreds	As much as possible

THANK YOU !

- Scan my face to get the slides !
- Check out the example testnet deployment :
 - [github.com/PierreBesson/
polkadot-kubernetes-testnet-example](https://github.com/PierreBesson/polkadot-kubernetes-testnet-example)
- Join our workshop at the Parachain Summit
(Nov 30 - Dec 1)



pierre-besson.fr/files/sub0-2022.pdf

Contact :

⌚ Element: @pierre.besson:matrix.parity.io

🐦 Twitter: @pibesson