cloud STACK COLLABORATION CONFERENCE

Extensions Framework & Orchestrate Anything

About Me

- **Harikrishna Patnala**
  - PMC and Committer @ Apache CloudStack project
  - Born and brought up in CloudStack
  - Lead Software Engineer @ ShapeBlue

Recap – CCC 2024

Why Extensions?

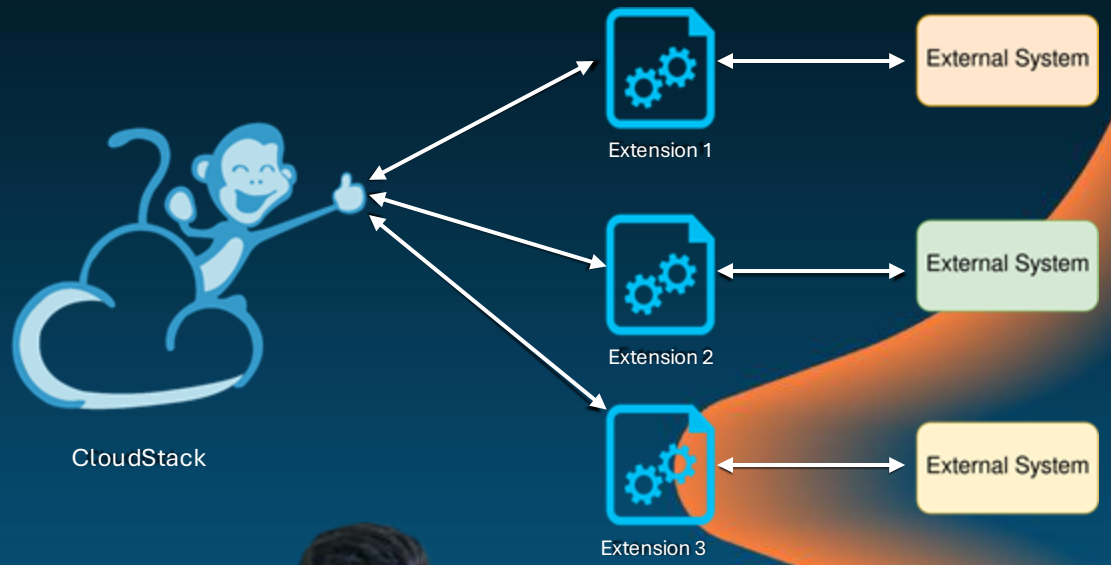
- Plug in External Logic Easily
  - Integrate custom scripts or tools directly into CloudStack workflows
  - Ideal for operators and developers outside the core project
- Decouples external logic from CloudStack core
- Integrations can be done easily

What can be Achieved?

- Can integrate new Instance provisioners or hypervisors
  - Proxmox
  - Hyper-V
  - Firecracker
  - MaaS
  - Baremetal
- Define custom actions like
  - Snapshots
  - Clone operations
  - Backups
- Can integrate new Network Extensions

Extensions Framework

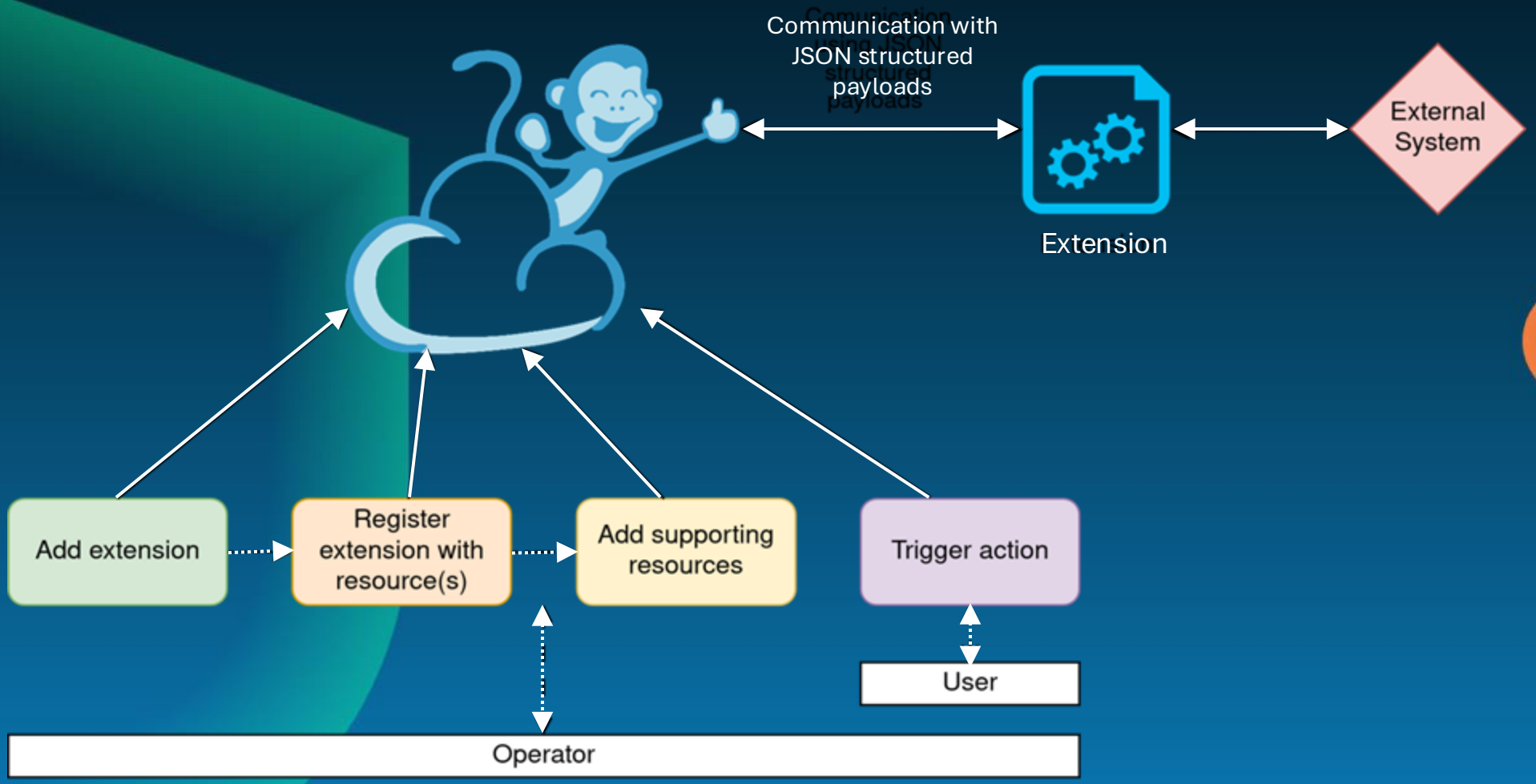
- Integrates external systems and workflows
- An executable binary or script in any programming language that acts as a bridge between CloudStack and the external system
- Available from 4.21.0



Extensions Framework

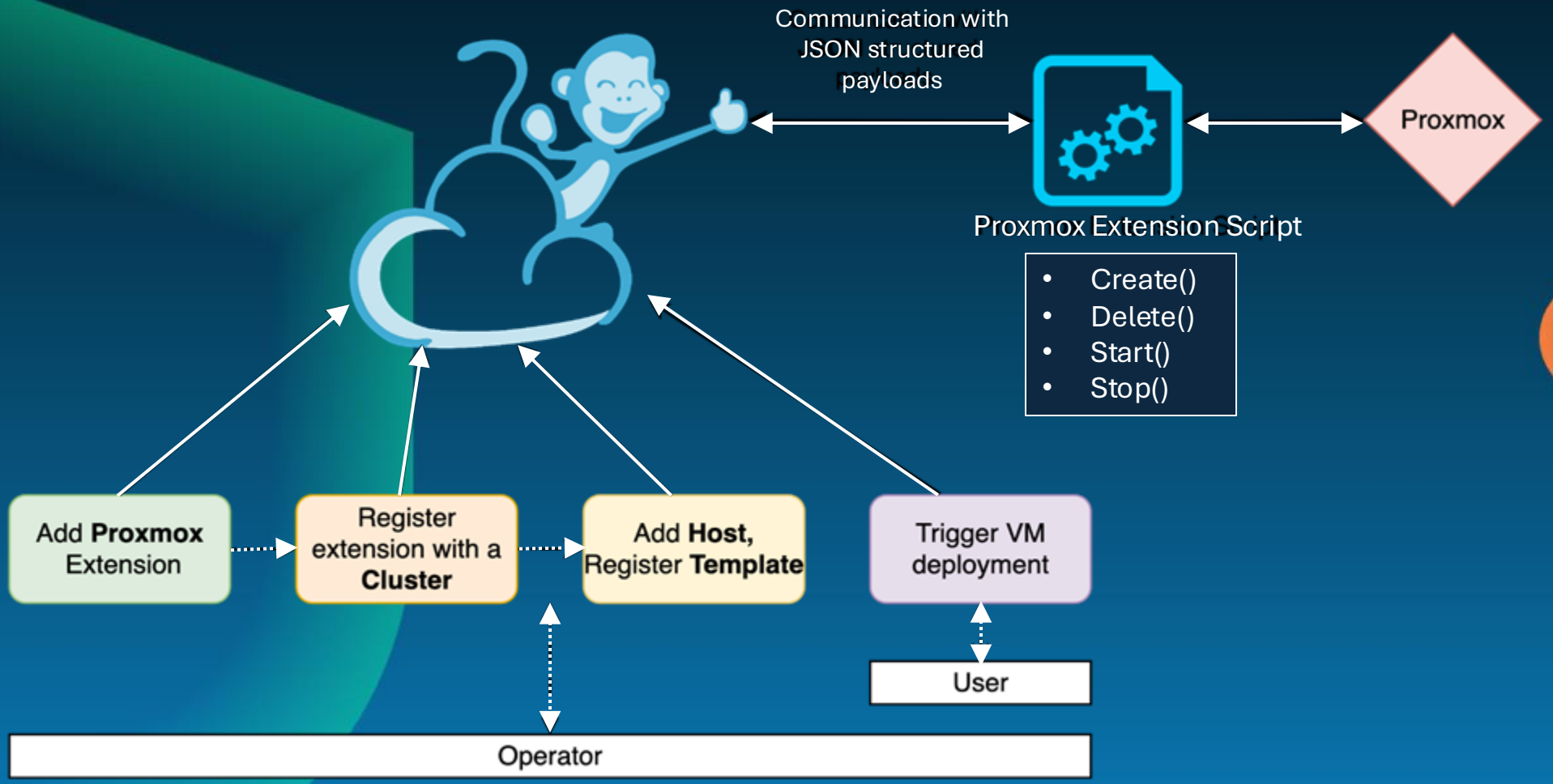
- Extensions of different types can be defined.
  - Currently **Orchestrator** type is supported.
- Communication using **JSON** structured payload
- Ability to define **custom actions**
- Extension **binary or script file**(s) will be placed at `/usr/share/cloudstack-management/extensions/<EXTENSION_NAME>`
- Extension data will be stored at `/var/lib/cloudstack/management/extensions/<EXTENSION_NAME>`

Extension Framework - Workflow



Extension Framework

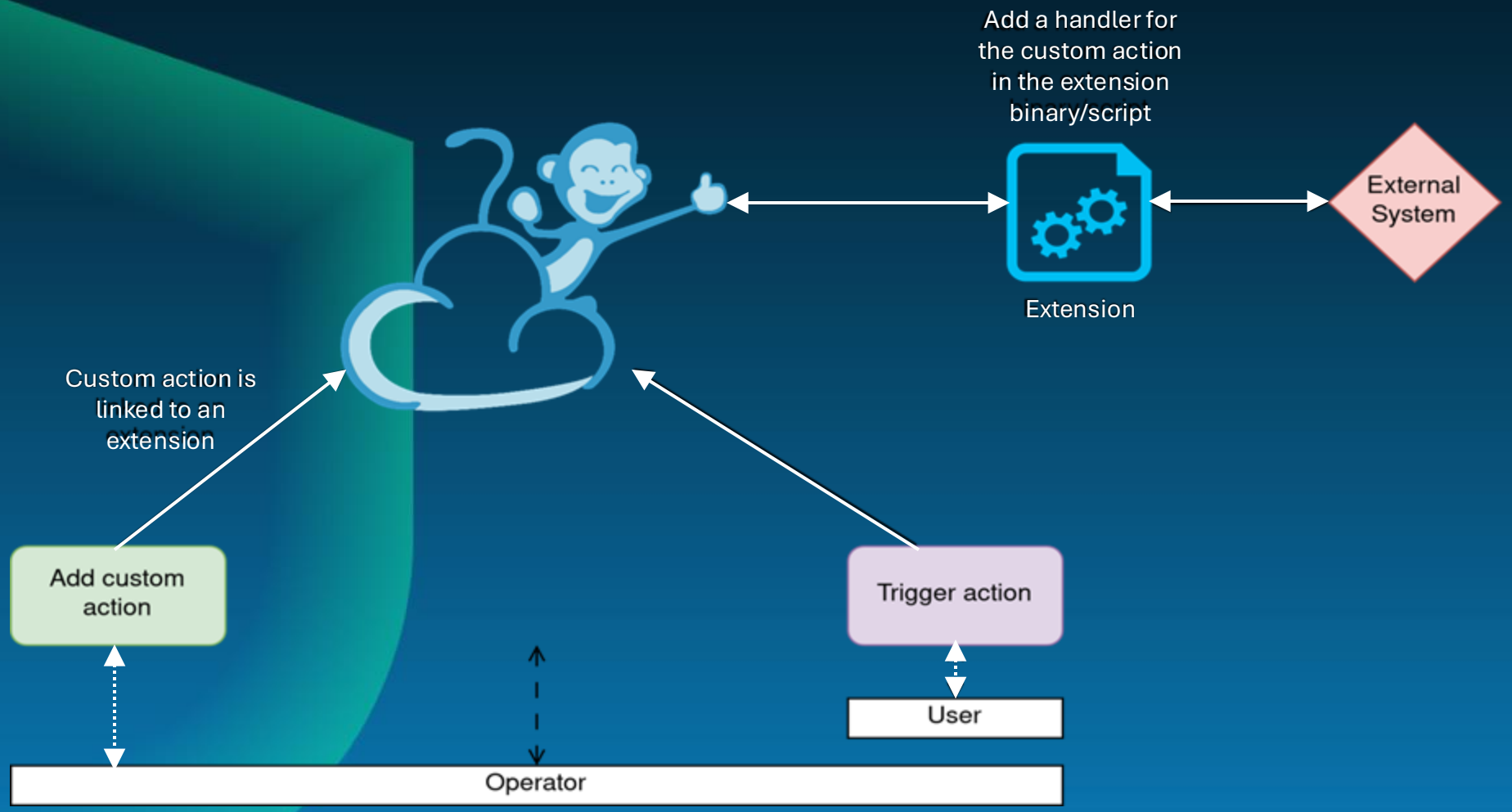
Workflow Example



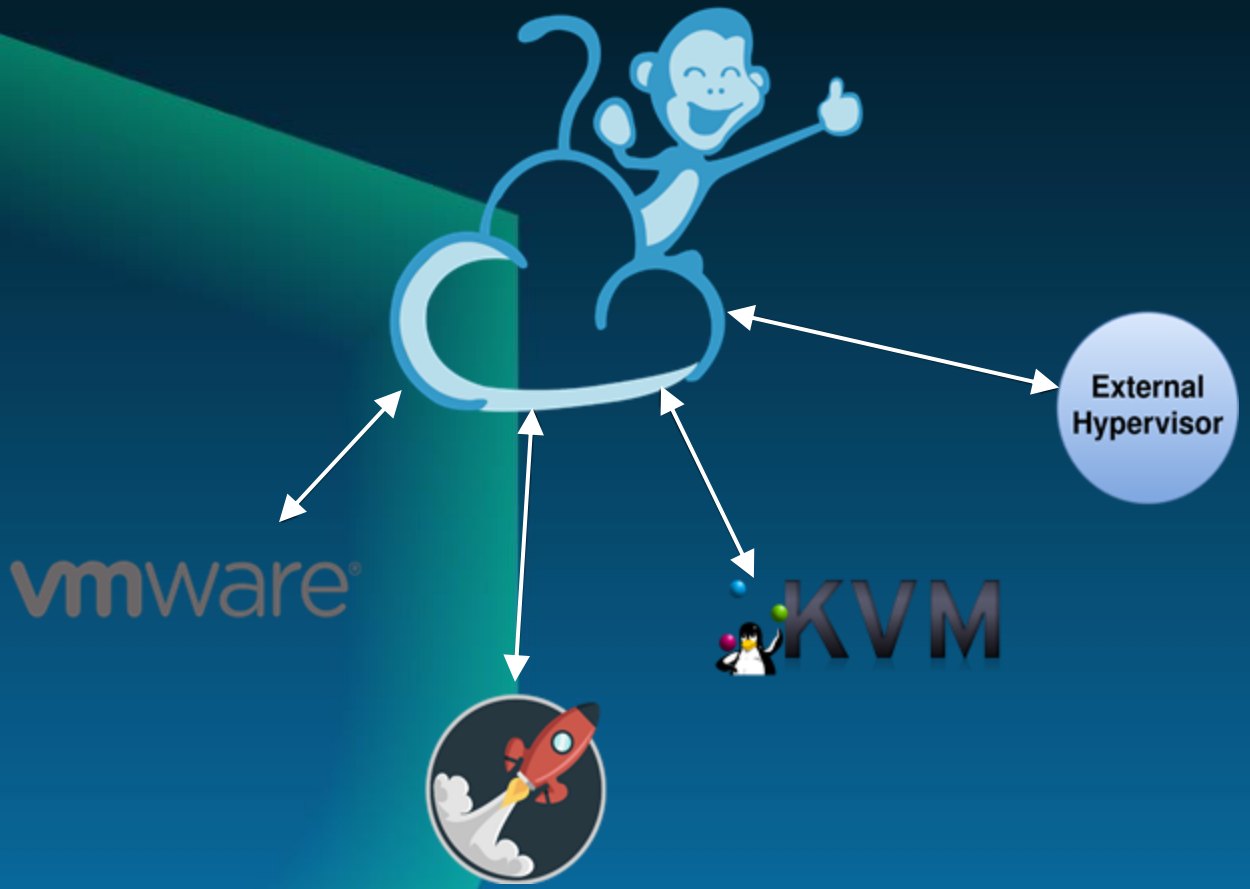
Custom Actions

- Operators can define custom actions for each extension
- Supports user-defined input parameters, success/error messages, allowed role types
- Actions can be linked to specific resource types

Custom Actions - Workflow



Orchestrator Extension



- Allows instance deployment on external systems
- Built-in extensions added for:
  - Proxmox
  - Hyper-V
  - MaaS
- Allows deploy, start, stop, reboot, expunge operations. More can be added using custom actions
- (Optional) Prepare action allows extension to update some of the fields CloudStack instance before deployment
  - Eg. MAC address for the Instance

Built-in Extensions

cloud STACK COLLABORATION CONFERENCE

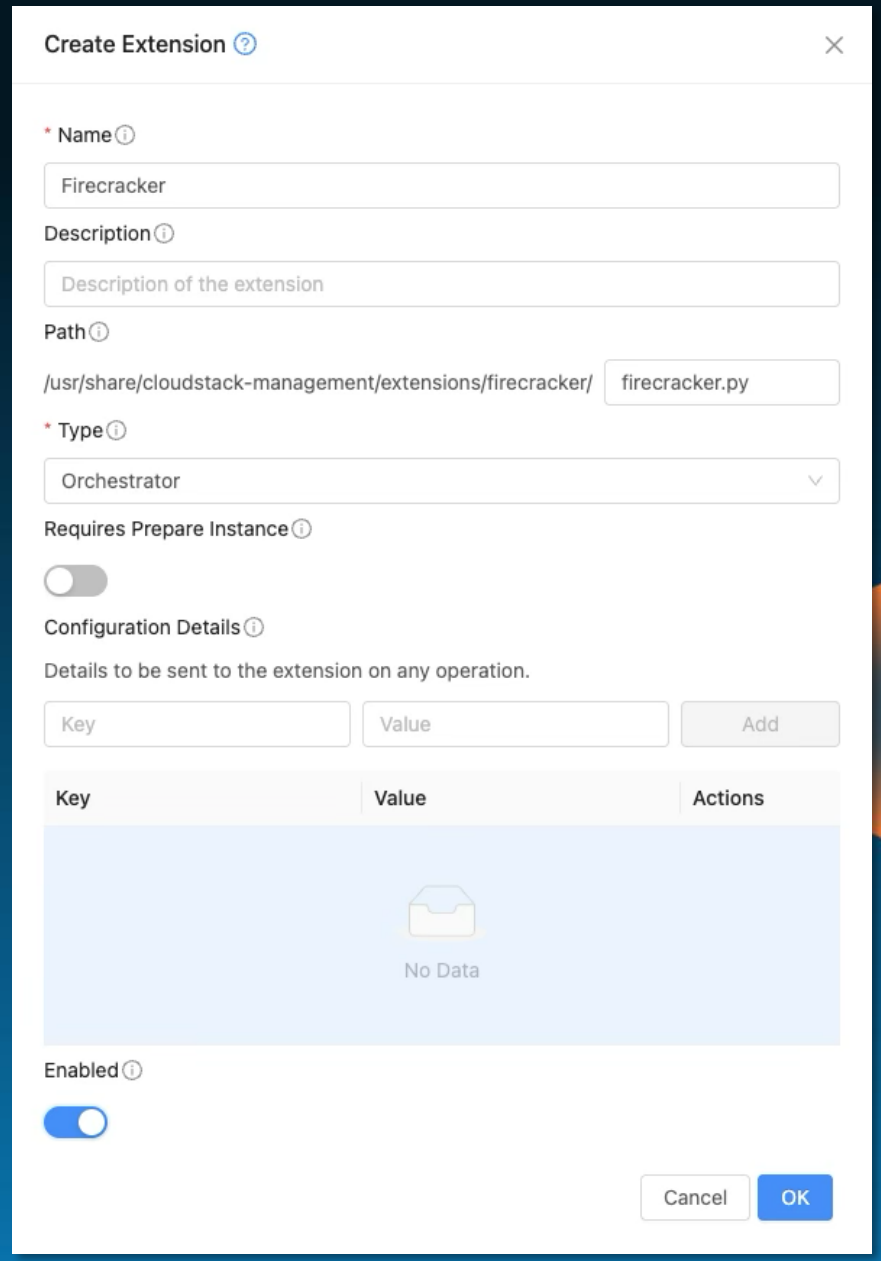
/ Extensions | Refresh | All | Create Extension + | Search

Name	State	Type	Path	Availability	Created
MaaS Inbuilt	Enabled	Orchestrator	/usr/share/cloudstack-management/extensions/MaaS/maas.py	Ready	11 Nov 2025 08:58:41
HyperV Inbuilt	Enabled	Orchestrator	/usr/share/cloudstack-management/extensions/HyperV/hyperv.py	Ready	11 Nov 2025 08:58:40
Proxmox Inbuilt	Enabled	Orchestrator	/usr/share/cloudstack-management/extensions/Proxmox/proxmox.sh	Ready	11 Nov 2025 08:58:40

Showing 1-4 of 4 items < 1 > 20 / page

Adding Extension

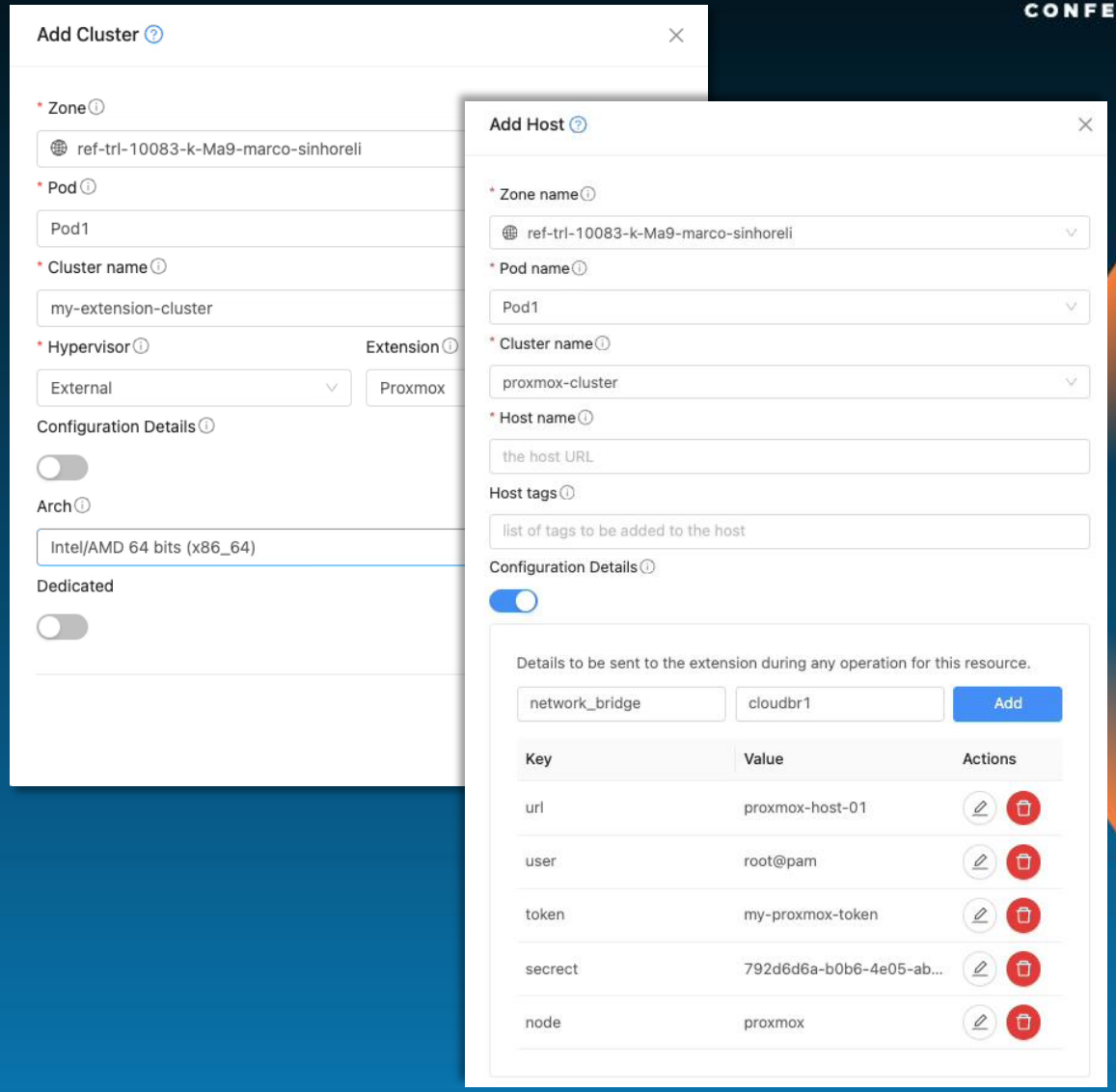
- Basic details:
  - name
  - path
  - type
  - type-specific configuration
- Optional metadata in form of key-value pair which will be passed to the binary/script



Registering Extension Resources

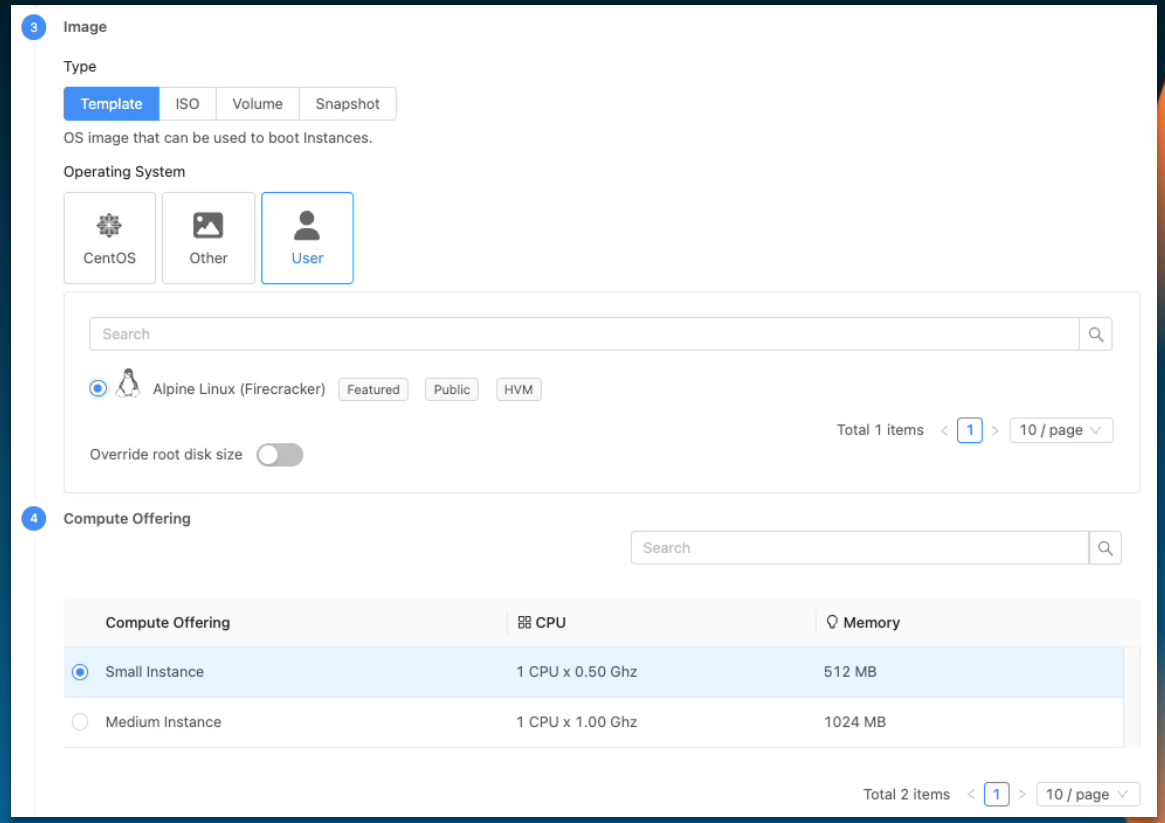
- For Orchestrator Type:
  - Clusters
  - Hosts
  - Templates
  - Service Offering (Optional)

- Optional metadata in form of key-value pair which will be passed to the binary/script



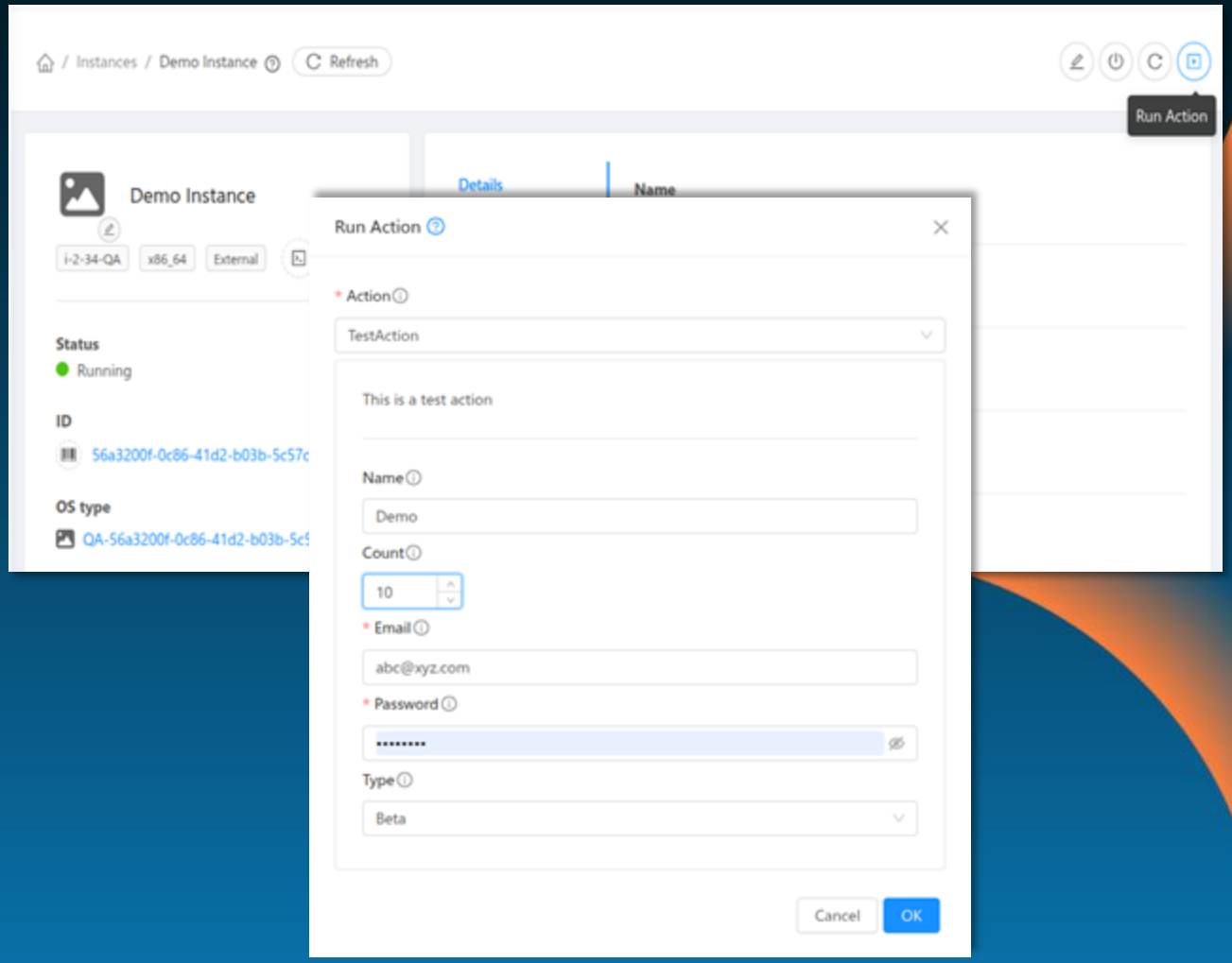
Using Extension (Trigger Action)

- No specific difference for Users
- For Orchestrator, Users will select the corresponding Template and the Instance will be deployed



Running Custom Action

- Run action show for the applicable resources
- Auto generated UI with value options, validations



cloud STACK COLLABORATION CONFERENCE

Firecracker Extension

About me

- **Marco Sinhoreli**

  - Committer @ Apache CloudStack Project
  - 25+ years in IT and cloud infrastructure
  - Technical Marketing Manager @ ShapeBlue

Proof of Concept – Use With Caution

The Firecracker extension is still experimental and has not been validated for production.

Testing in non-critical environments is strongly recommended.

About Firecracker

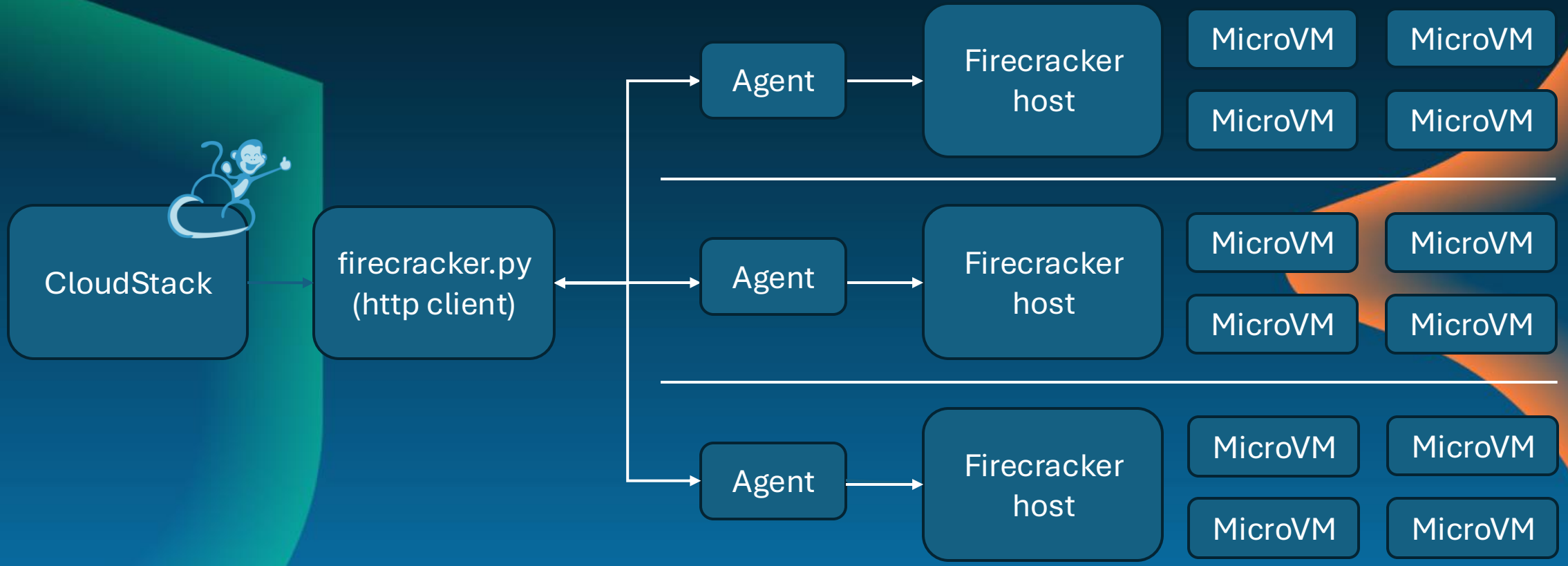
Lightweight VMM by AWS for serverless, edge, and container workloads

Boots microVMs in milliseconds with minimal memory overhead

Limitation: lacks orchestration and high-level REST APIs

Goal: let CloudStack manage Firecracker microVMs as standard Instances

Firecracker Extension + Agent



Firecracker Agent Capabilities

cloud STACK COLLABORATION CONFERENCE

- REST API endpoints: /v1/create, /v1/stop, /v1/delete
- Storage backends: file, LVM, LVM-thin
- Networking backends: Linux bridge (VLAN-aware) and OVS
- Process control via tmux for isolation and restart recovery
- Idempotent operations with persistent state on the Host
- AuthN/AuthZ via PAM
- TLS + mTLS encryption for all endpoints

cloud STACK COLLABORATION CONFERENCE

Demo

Future & What's Next

- New Types - Network, Authenticator, etc
- Usability Improvements - feedback from community
- Extension marketplace?

cloud STACK COLLABORATION CONFERENCE

QA

Resources

- https://docs.cloudstack.apache.org/en/latest/adminguide/extensions.html
- https://docs.cloudstack.apache.org/en/latest/adminguide/extensions/inbuilt_extensions.html
- https://github.com/msinhore/cloudstack-firecracker-extension