cloud STACK | COLLABORATION CONFERENCE

External Compute Deployment in Apache CloudStack

Harikrishna Patnala
Alex Mattioli

November 20 - 22, 2024 | Madrid, Spain

Who we are

Harikrishna Patnala:

- from Hyderabad, India
- Software Development Engineer at ShapeBlue
- PMC member and Committer in Apache CloudStack
- Born and raised in CloudStack
- Previously worked at Accelerite and Citrix

Alex Mattioli:

- From everywhere
- 30 years in IT, 20 in Infrastructure
- Cloud Architect at ShapeBlue
- Involved in CloudStack since 2012
- Built some very large clouds with ACS
- Architected many ACS features
- Apache CloudStack Committer

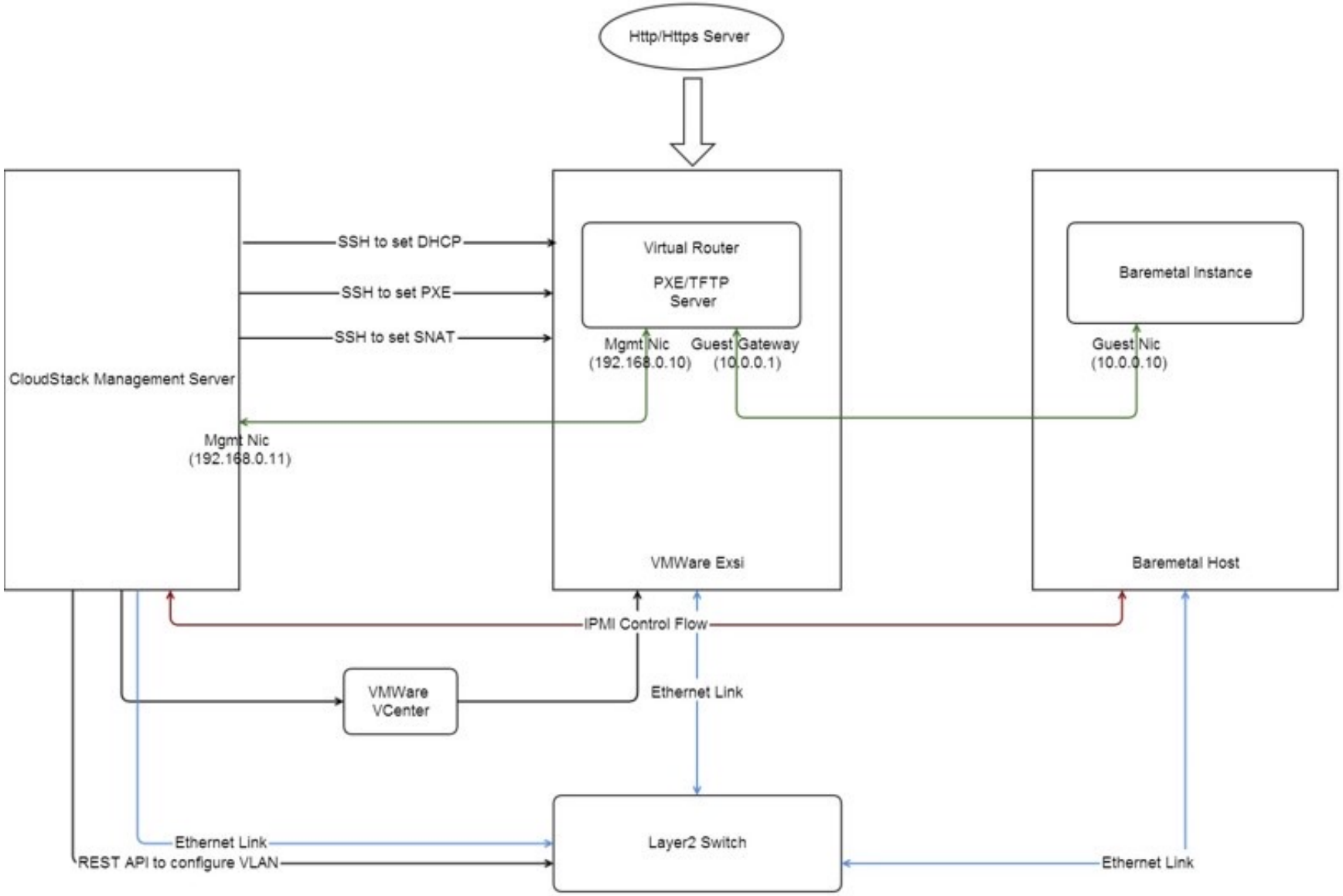
External Compute Resource Framework

Community

4.21

- Code Complete
- Fully Tested

Baremetal
Old
Way



Dell Force10

External
Compute
Resources

External Compute Resource

Any compute resource that isn’t natively controlled by Apache CloudStack.

External Compute Resource

- VMs in a non-supported hypervisor
- **Baremetal servers**
- Serverless platforms
- Applications
- Database Cluster
- IP-PBX/SBCs
- HPC Clusters
- Shiv's Robot*
- Edge and IOT devices
- Network Appliances
- Cloud Resources from External Providers

*Explanations after the talk

External Compute Resource

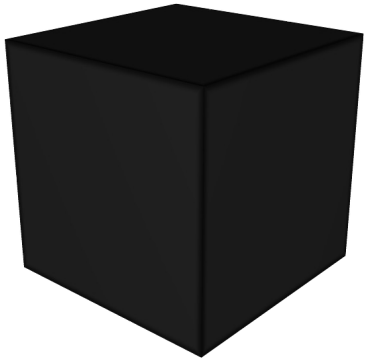
Anything that computes which can be deployed and managed via an API or CLI

Deployment

External Compute Resource Deployment

Provisioner Script

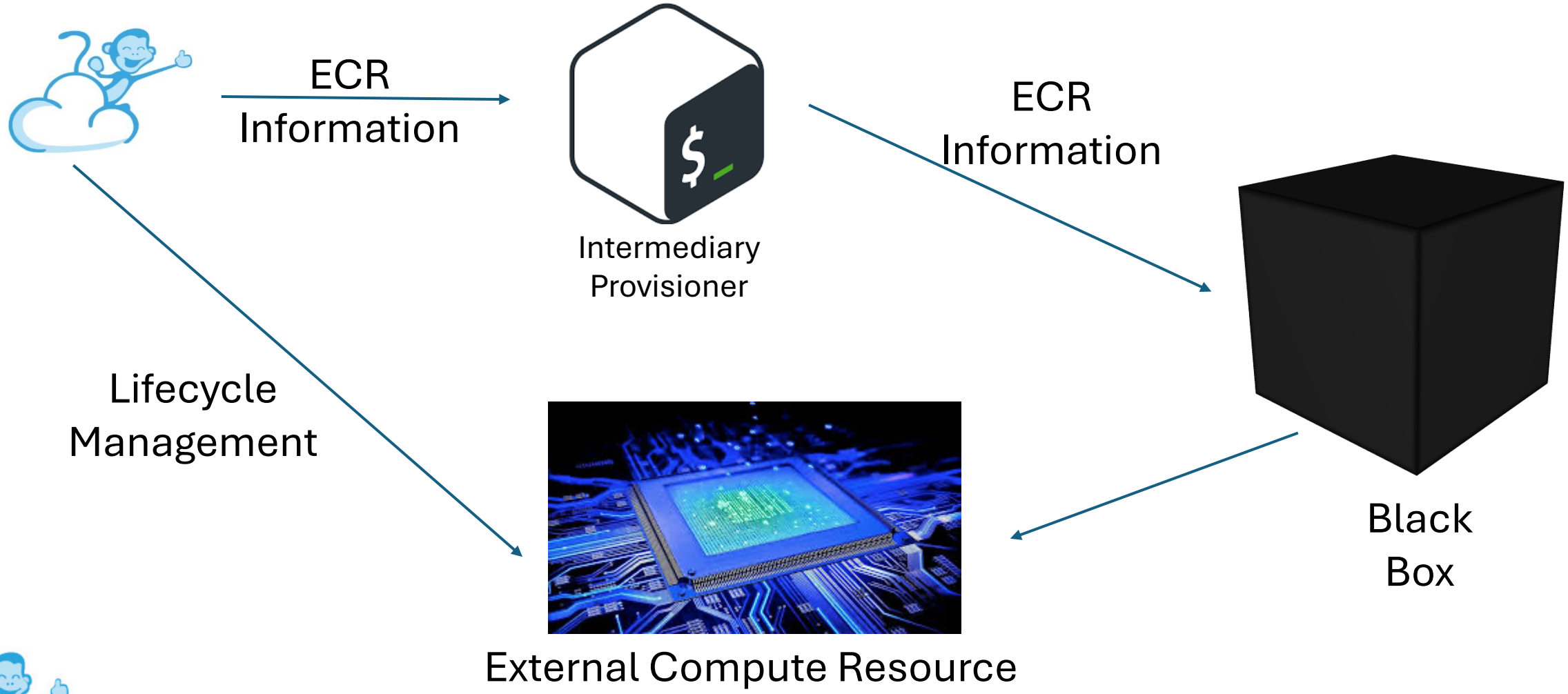
- Resides in ACS
- BASH Script
- Middleware - ACS <-> Black box
- Sends Requests to BlackBox



Black Box

- External to ACS
- Any API endpoint
- Receives request from Provisioner Script
- BlackBox from ACS's POV

External Compute Resource Deployment



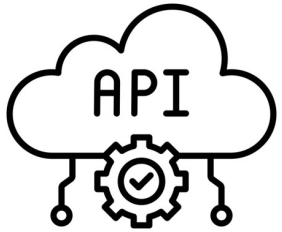
Baremetal New Way

Baremetal Deployment

Offering
Template
Vlan

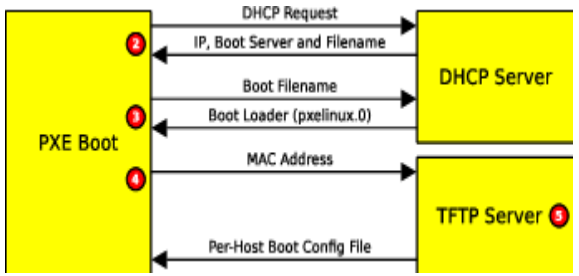
ACS Bare-metal Provisioner

Offering
Template
Vlan



Bare-metal Deployment API

Lifecycle Management



Baremetal Deployment Pipeline

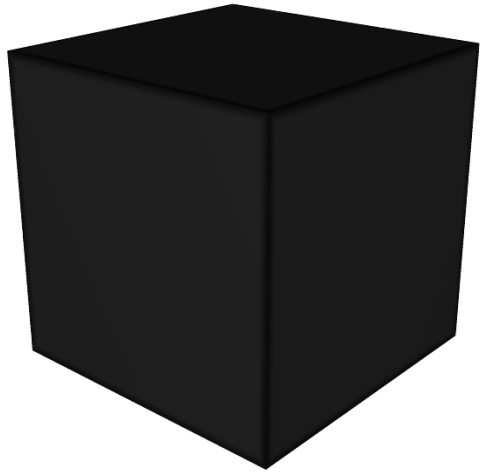
Architect's Simple Example(s)

Architect’s Simple Example: Proxmox

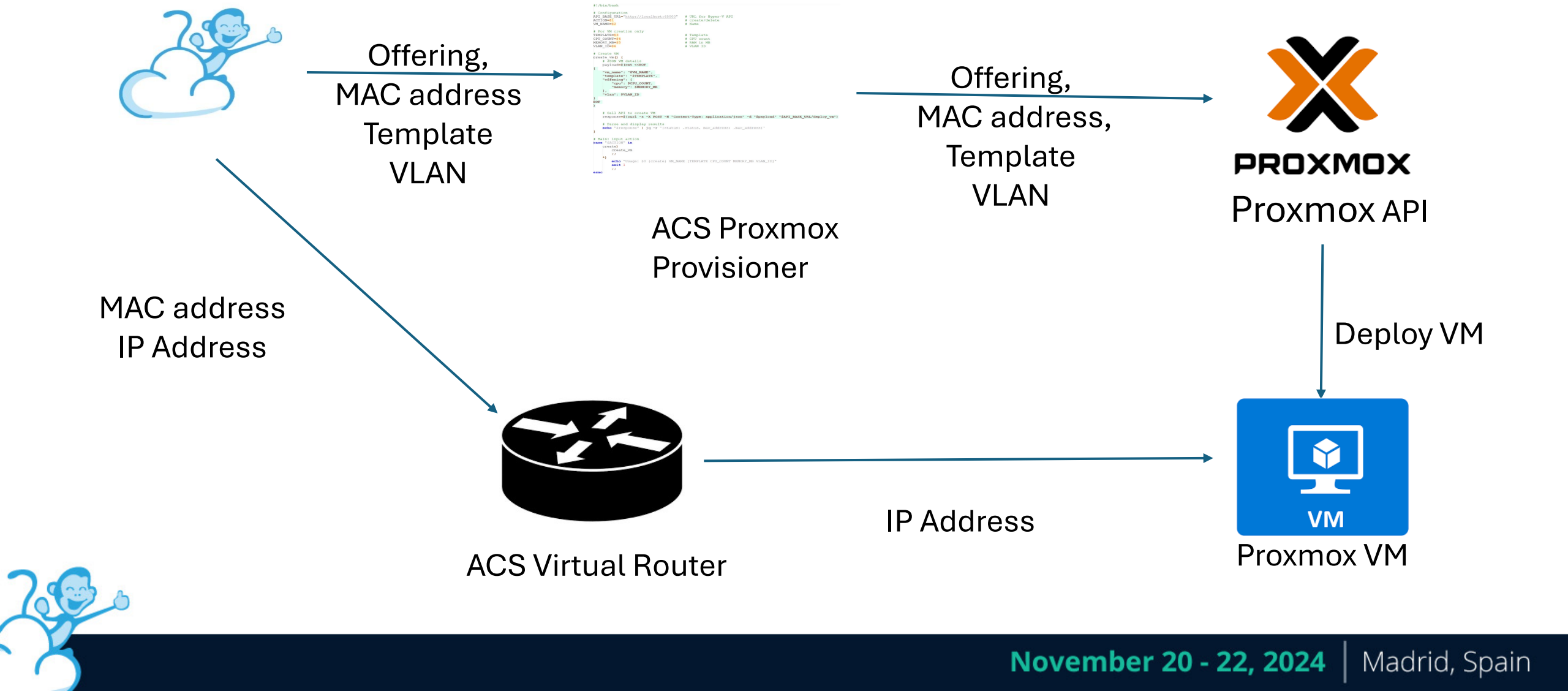
Proxmox provisioner: simple BASH script

```bash
#!/bin/bash

# Configuration
API_BASE_URL="http://localhost:65000"   # URL for Hyper-V API
CLOUDSTACK_API_URL="http://cloudstack-server:8080/client/api"  # URL for CloudStack API
API_KEY="your-api-key"
SECRET_KEY="your-secret-key"

VM_NAME=$1                              # VM Name
OFFERING_NAME=$2                        # Offering

# Get CPU and RAM from offering
get_offering_details() {
    response=$(curl -s "$CLOUDSTACK_API_URL?command=listServiceOfferings&apiKey=$API_KEY&secretKey=$SECRET_KEY")

    # Extract CPU and RAM
    cpu_count=$(echo $response | jq -r ".listserviceofferings.serviceoffering[?(.name==\"$OFFERING_NAME\")].cpunumber")
    memory_mb=$(echo $response | jq -r ".listserviceofferings.serviceoffering[?(.name==\"$OFFERING_NAME\")].memory")
}

TEMPLATE=$3                             # Template
VLAN_ID=$4                              # VLAN ID

# Create VM
create_vm() {
    get_offering_details

    # JSON VM details
    payload=$(cat <<EOF
{
    "vm_name": "$VM_NAME",
    "template": "$TEMPLATE",
    "offering": {
        "cpu": $cpu_count,
        "memory": $memory_mb
    },
    "vlan": $VLAN_ID
}
EOF
)

    # Call Hyper-V API -> create VM
    response=$(curl -s -X POST -H "Content-Type: application/json" -d "$payload" "$API_BASE_URL/deploy_vm")

    # Status and Fetch MAC Address
    echo "$response" | jq -r '{status: .status, mac_address: .mac_address}'
}

#Main
create vm
```

Proxmox API



Architect’s Simple Example: Proxmox

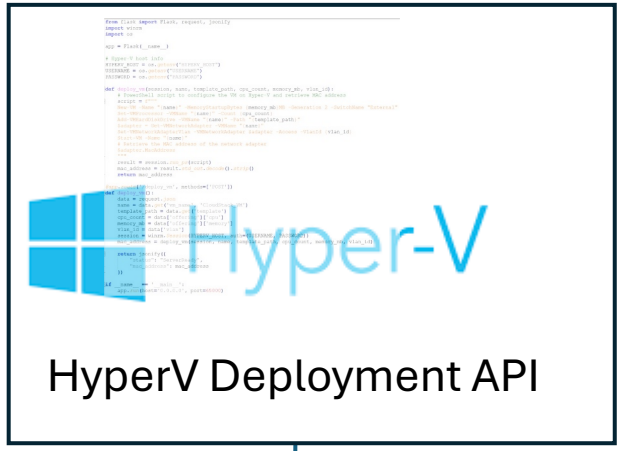


Another Possibility: HyperV (with middleware)

Offering
Template
Vlan

ACS HyperV Provisioner

Offering
Template
Vlan

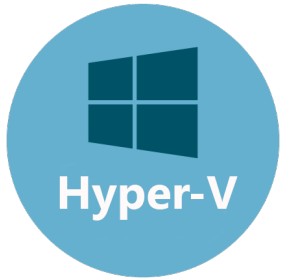


HyperV Deployment API

Template
path
CPU #
Memory #
VLAN

PowerShell
WinRM

Deploy VM



VM

Code is available for
anyone interested.

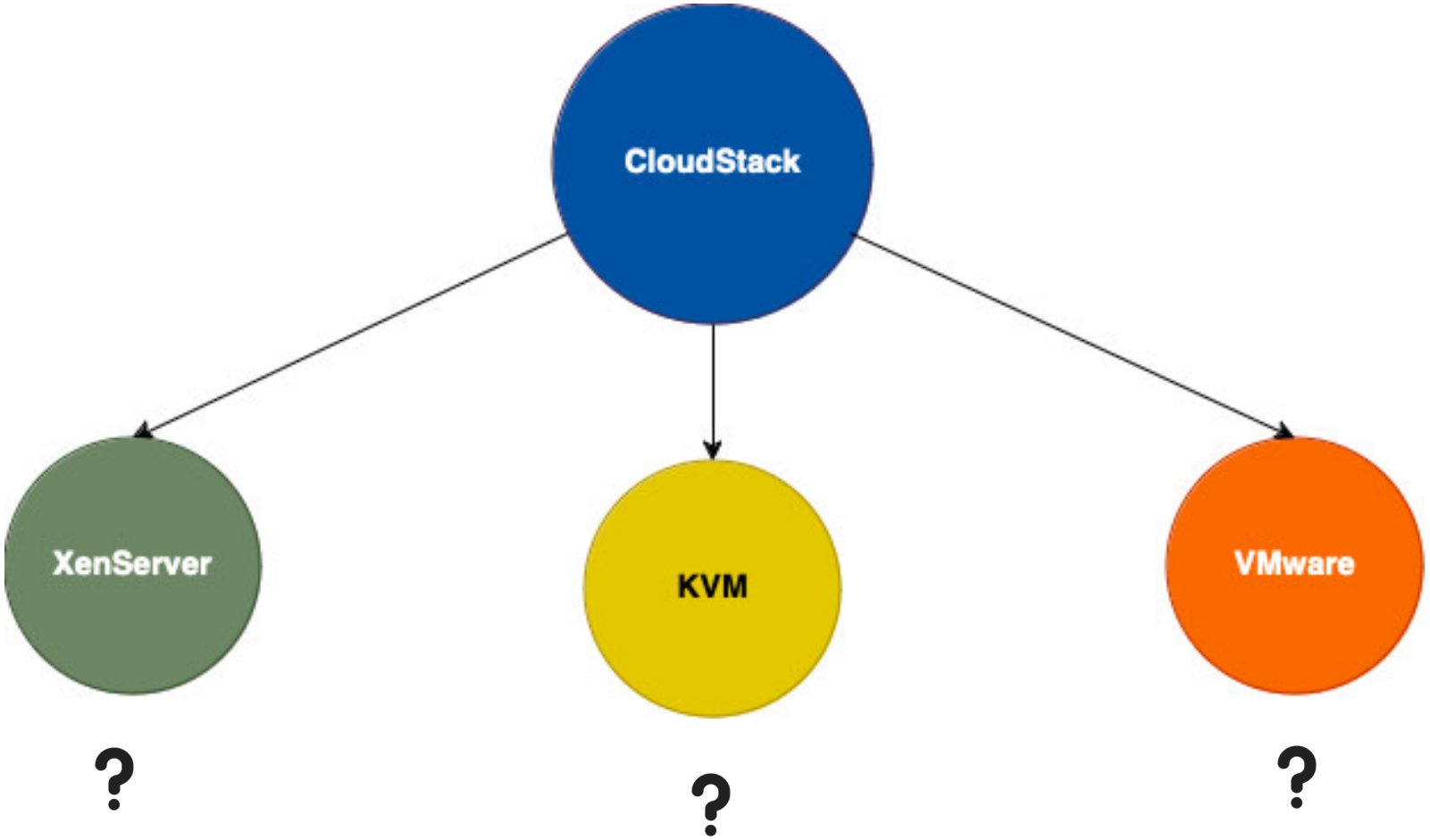
Over

To

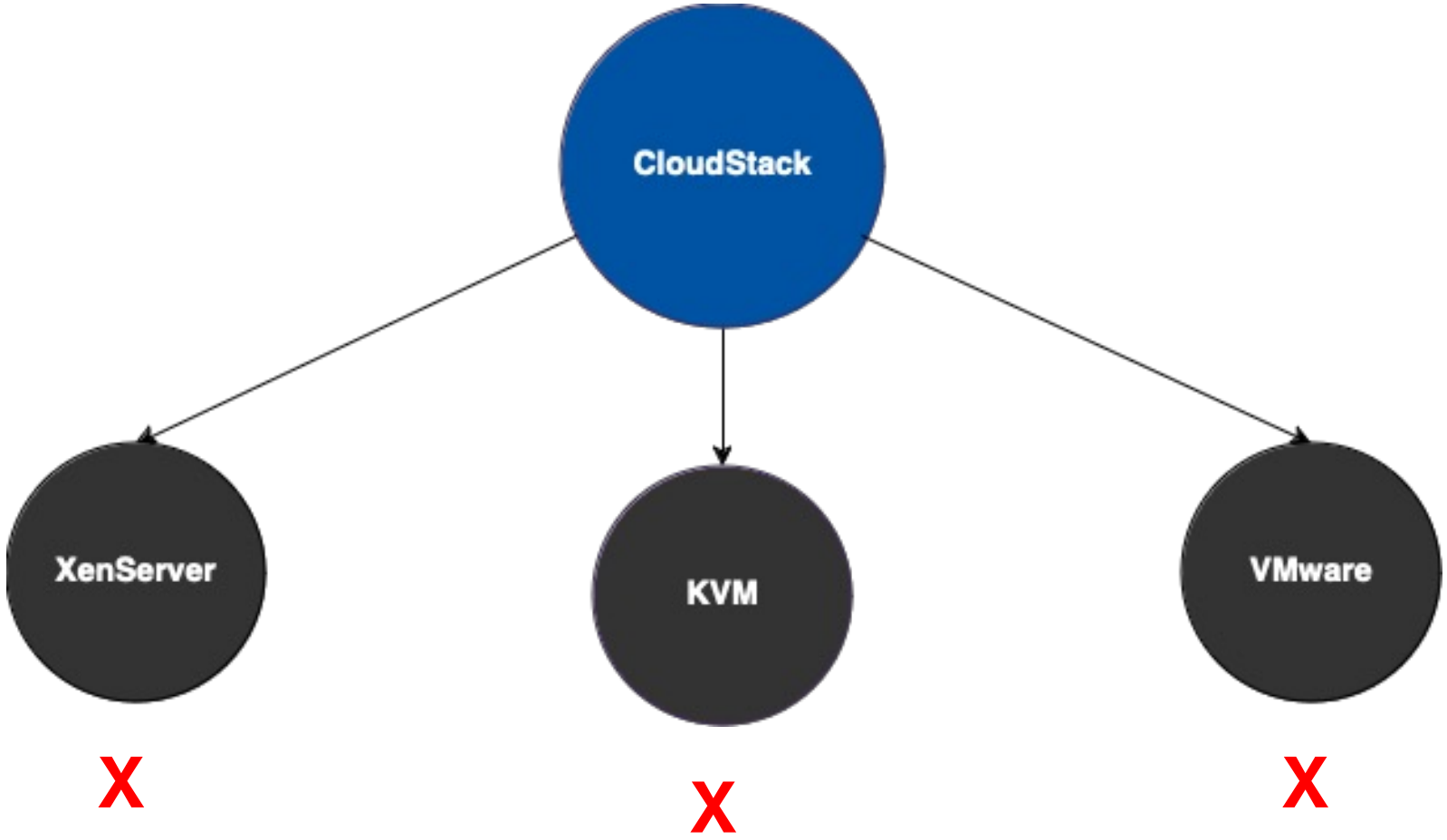
Hari

Implementation

How does this fit in CloudStack ?



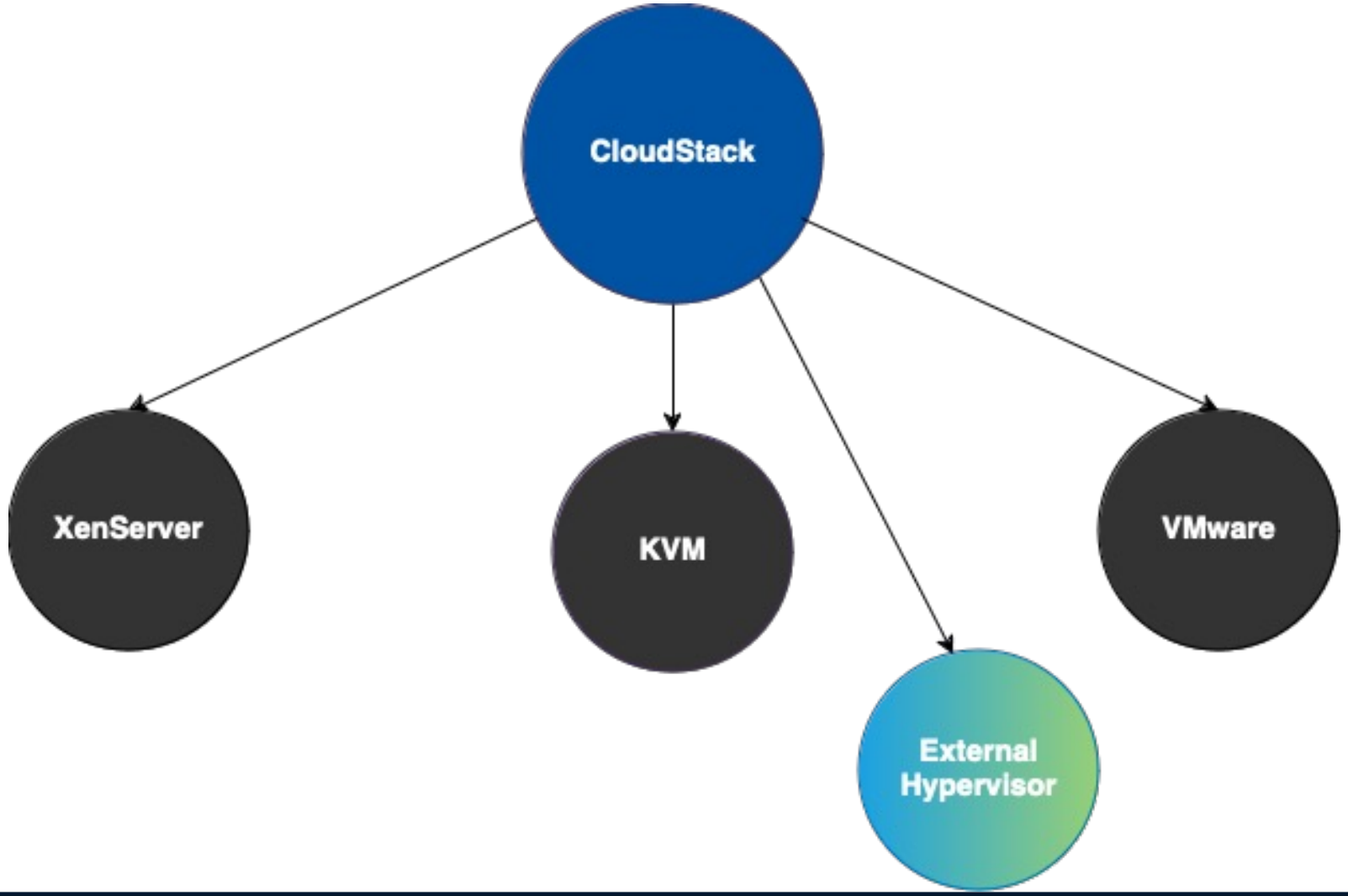
Does not fit in the existing hypervisor model



Fitting ECS into CloudStack

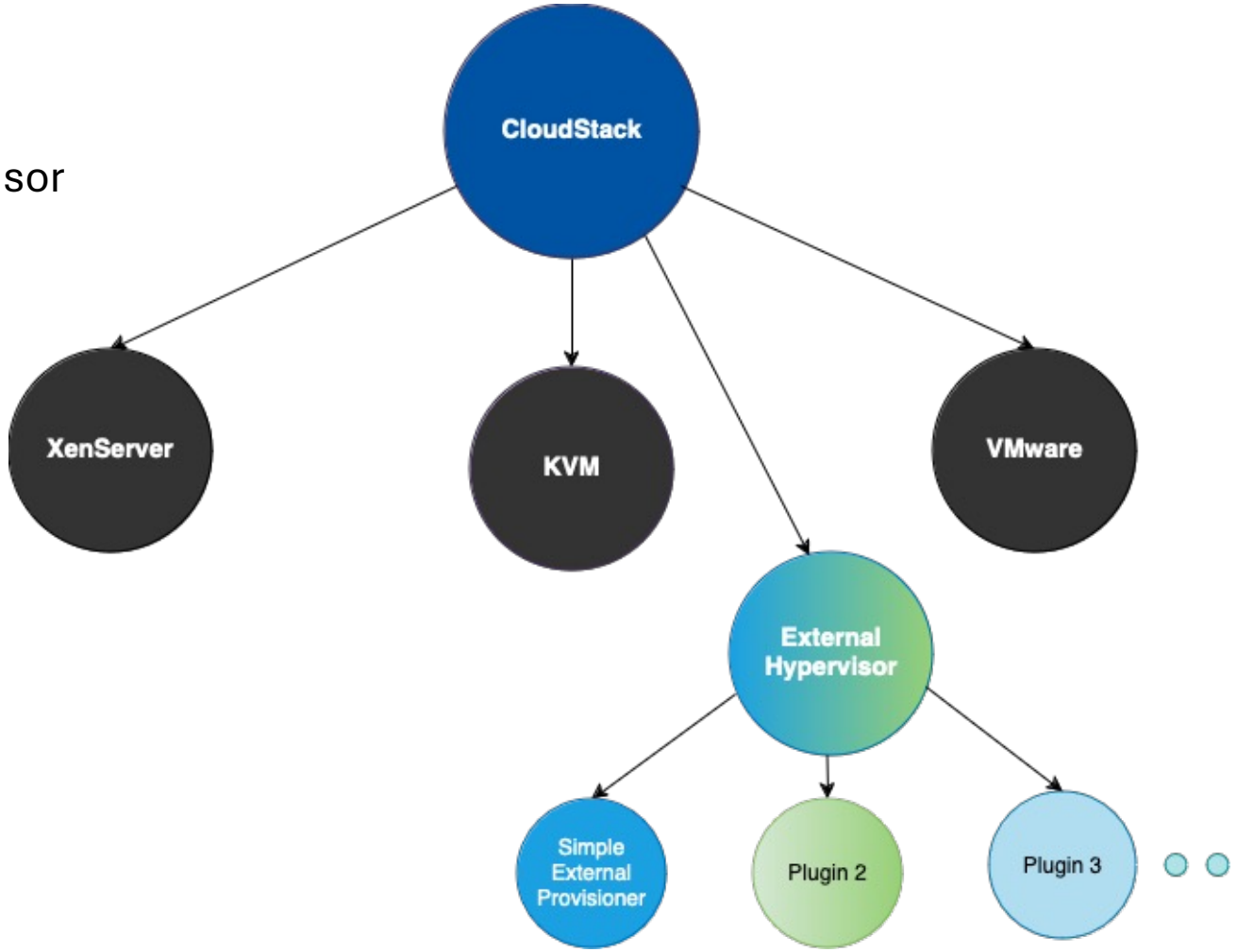
- CloudStack understands hypervisors
- CloudStack POV -> External Provisioner = hypervisor

Introducing External Compute Resource



Plugin model

- Support for multiple plugins under External Hypervisor
- Added a default “Simple External Provisioner”



Simple External Provisioner Scripts

Simple
Old
BASH

provisioner.sh powerOperations.sh

```
36
37  prepare() {
38      parsed_arguments=$(parse_json "$1")
39      mac_address=$(generate_random_mac)
40      mac_json=$(jq -n --arg mac "$mac_address" '{"mac_address": $mac}')
41      echo "$mac_json"
42
43      # Add code to handle preparation logic
44  }
45
46  create() {
47      parsed_arguments=$(parse_json "$1")
48      # Add code to handle creation logic
49  }
50
51  delete() {
52      parsed_arguments=$(parse_json "$1")
53      # Add code to handle delete logic
54  }
55
```

Workflow In CloudStack

apachecloudstack™

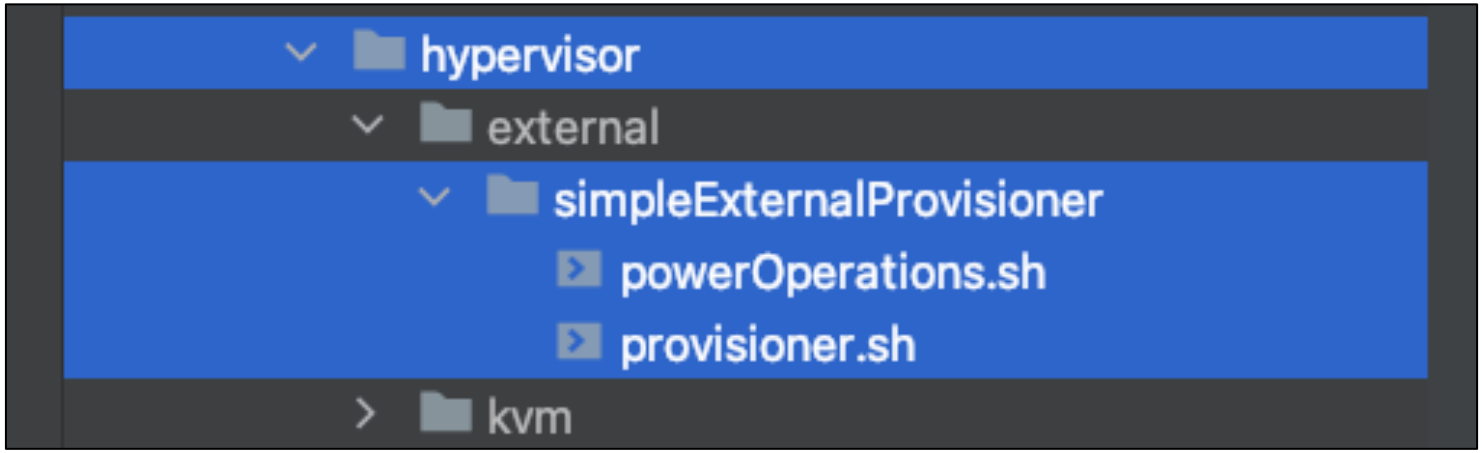
Plugin configuration

Global / Cluster settings -

External provisioners (external.provisioners)
List of external hypervisor provisioners

simpleExternalProvisioner

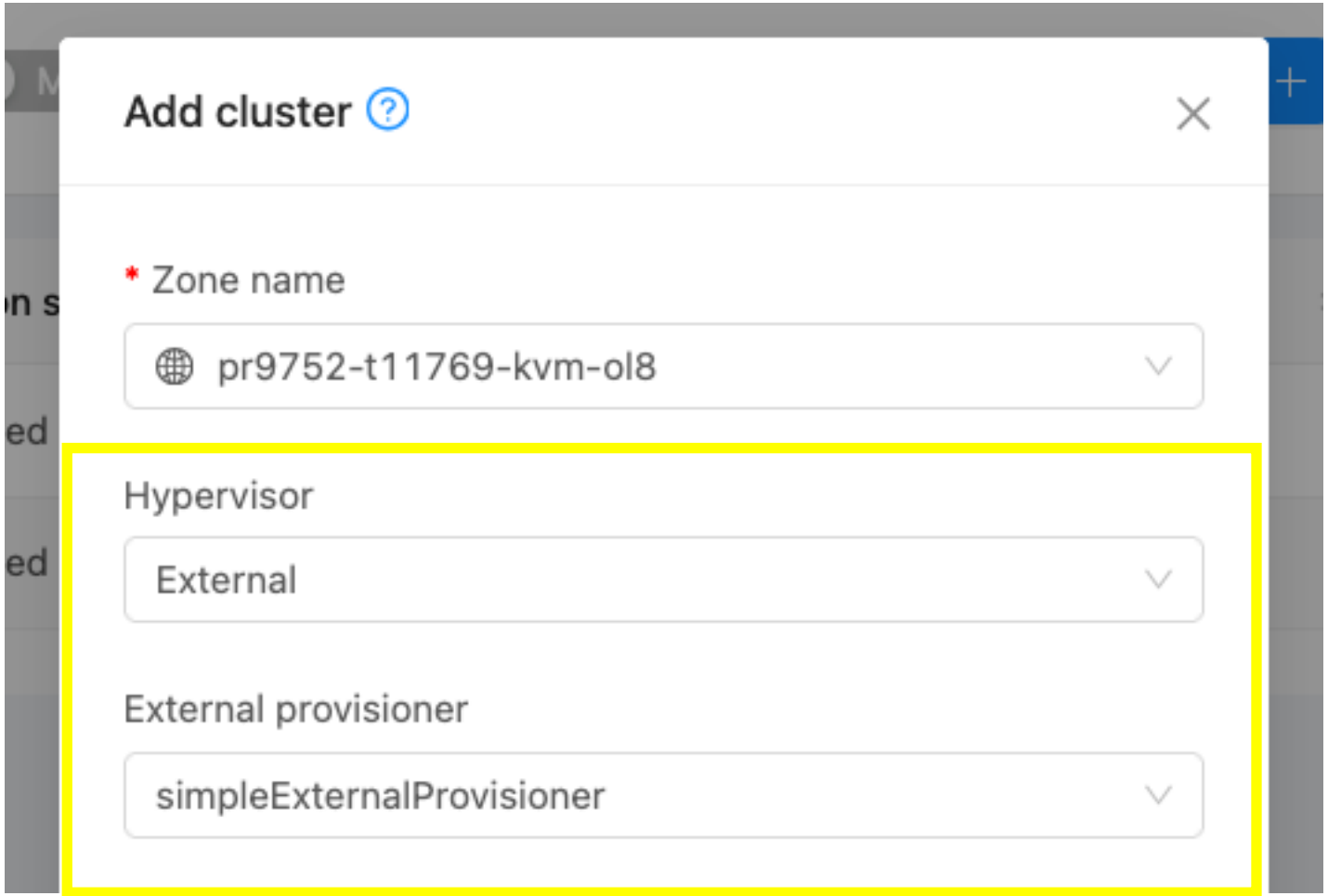
Expect macaddress from external provisioner
(expect.macaddress.from.external.provisioner)
Sample external provisioning config, any value that has to be sent



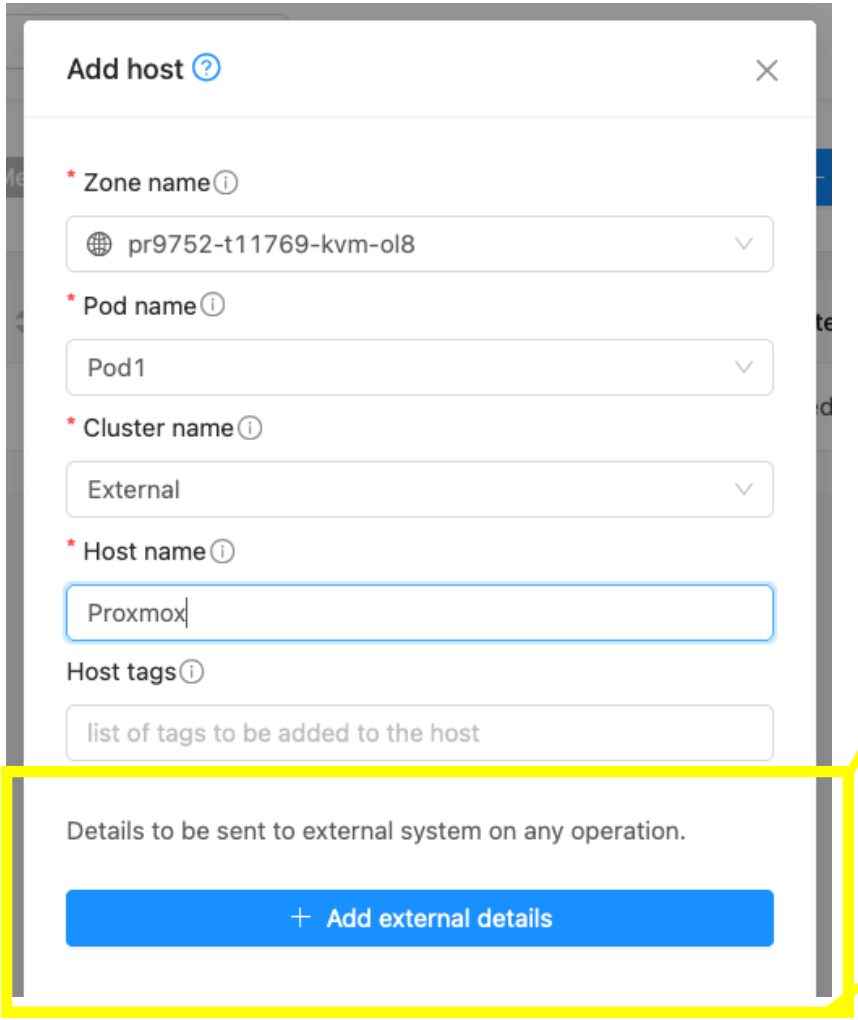
- BlackBox Scripts

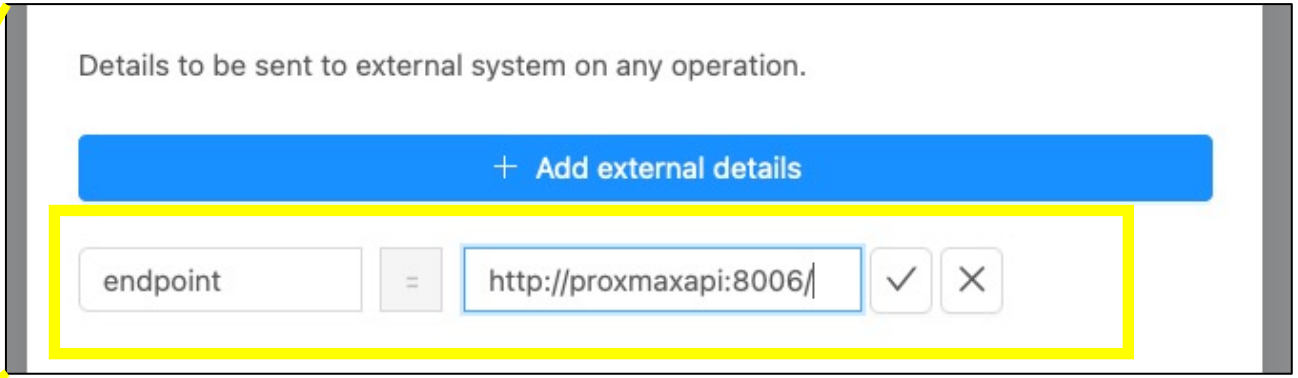
Location on MS: /usr/share/cloudstack-common/scripts/vm/hypervisor/external/simpleExternalProvisioner/

Adding Cluster

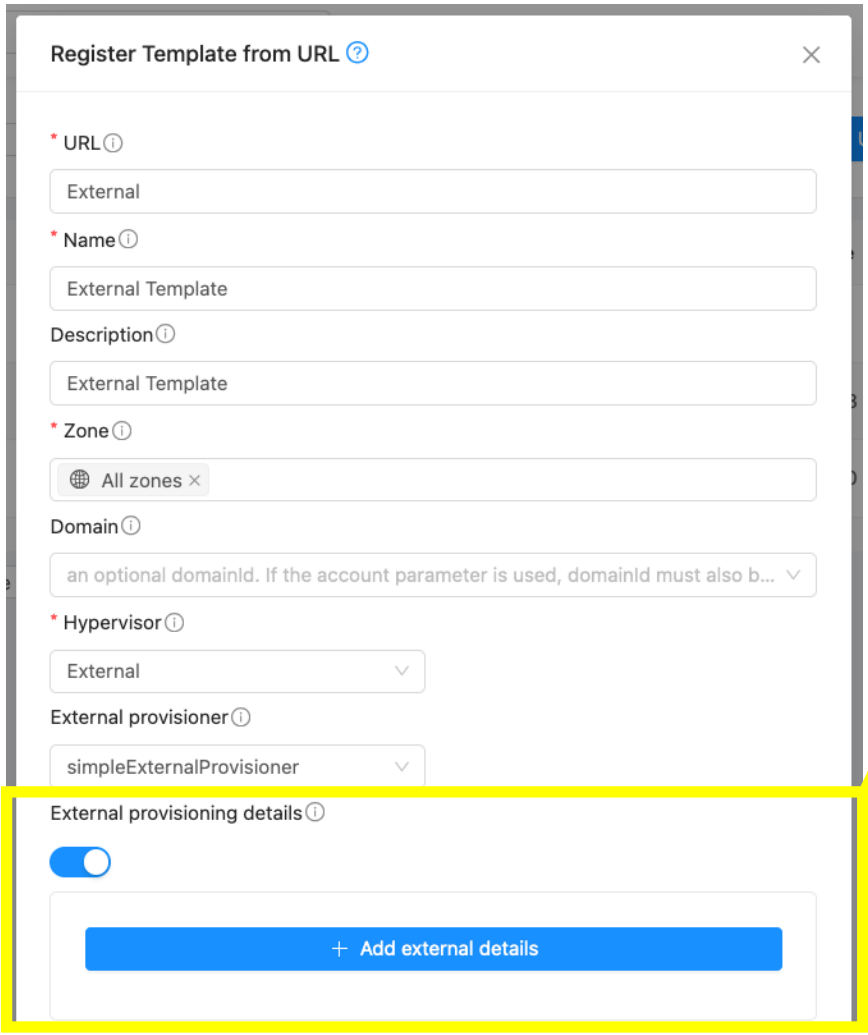


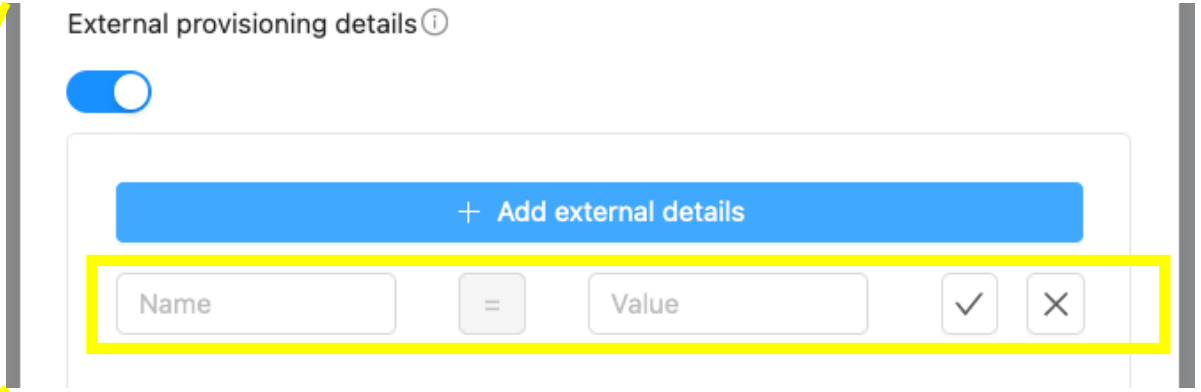
Adding ECR host



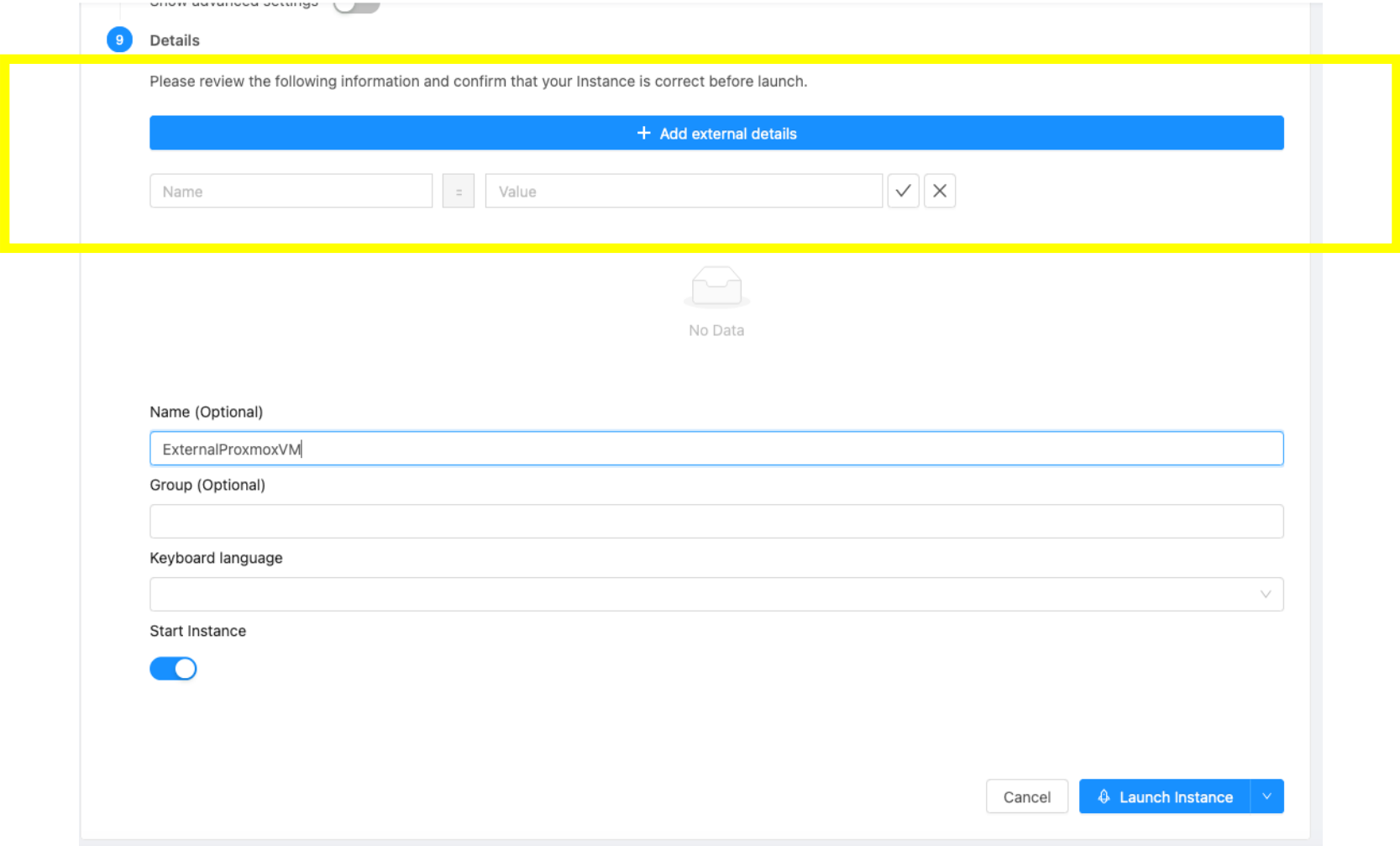


Adding ECR template





Deploy VM


9
Details
Please review the following information and confirm that your Instance is correct before launch.
+ Add external details
Name
=
Value
No Data
Name (Optional)
ExternalProxmoxVM
Group (Optional)
Keyboard language
Start Instance
Cancel
Launch Instance

DEMO


STAKEHOLDERS
ME WATCHING MY SELLOFF
GO HORRIBLY EVEN THOUGH
I SAW IT WORK IN PRIVATE

References

- Github PR: https://github.com/apache/cloudstack/pull/9752
- Design document: https://cwiki.apache.org/confluence/display/CLOUDSTACK/External+Deployment+Integration+in+CloudStack

Questions?

Thank you

#CSCollab24
@CloudStack