ONEedge.io

A Software-defined Edge Computing Solution

D4.2. Infrastructure Report - b

Infrastructure Incremental Report

Version 1.0

10 March 2021

Abstract

This report, delivered at the end of the Second Innovation Cycle (M10-M16), describes in detail the architectural and operational specification of the highly distributed cloud reference infrastructure for edge computing designed and implemented in the ONEedge project. This edge cloud reference infrastructure is being used in the demonstration, testing and certification processes described in the first version of this deliverable (D4.1. “Infrastructure Report”). This report also includes relevant performance metrics and a detailed list of the tests and extensions implemented to verify the functionality of the software developed during the cycle for each component and software requirement.

This project has received funding from the European Union’s Horizon 2020 Research and Innovation Programme under Grant Agreement No 880412.

Deliverable Metadata

Project Title:	A Software-defined Edge Computing Solution
Project Acronym:	ONEedge
Call:	H2020-SMEInst-2018-2020-2
Grant Agreement:	880412
WP number and Title:	WP4. Demo and Operational Infrastructure
Nature:	R: Report
Dissemination Level:	PU: Public
Version:	1.0
Contractual Date of Delivery:	28/2/2021
Actual Date of Delivery:	10/3/2021
Lead Authors:	Vlastimil Holer, Rubén S. Montero and Constantino Vázquez
Authors:	Sergio Betanzos, Ricardo Díaz, Jim Freeman, Christian González, Alejandro Huertas, Shivang Kapoor, Jorge M. Lobo, Jan Orel and Petr Ospaly
Status:	Submitted

Document History

Version	Issue Date	Status[1]	Content and changes
1.0	10/3/2021	Submitted	First final version of the D.4.2 report

[1] A deliverable can be in one of these stages: Draft, Peer-Reviewed, Submitted and Approved.

Executive Summary

This report discusses the infrastructure specifications and the tools and services needed to set up a highly distributed cloud infrastructure for edge computing, including the foundation edge clusters and the front-end. It also covers the configuration details that are needed when the on-premises infrastructure grows with edge resources from bare-metal cloud providers, mainly Amazon Web Services (AWS) and Packet / Equinix Metal.

The performance improvements in ONEedge are measured in terms of performance and scalability of its components using the following metrics: Edge Cluster Deployment Performance, Application Deployment Performance, Application I/O Performance and Application Snapshot Performance. Our aim is to keep those metrics while new features are incorporated into the solution.

For each software requirement, this report also includes a detailed list of the extensions implemented to verify the functionality of the software developed in ONEedge during the Second Innovation Cycle (M10-M16). Also, for each requirement, we list the verification scenarios that have been addressed and a description of the functionality tested to fulfill the proposed scenarios. These extensions are used in the certification process, which was described in the first version of this deliverable (D4.1. “Infrastructure Report”), on the edge cloud reference infrastructure.

During the Second Innovation Cycle (M10-M16), the project mostly focused on those software requirements needed to achieve our second milestone in M16, which is the base functionality needed for a multi-host edge deployment. The work carried out during this Second Innovation Cycle involved software requirements from components CPNT1, CPNT2, CPNT3, CPNT4 and CPNT5, with a special focus on the edge instance management (CPNT1) and the deployment and provision of edge infrastructures (CPNT4).

Table of Contents

1. Edge Cloud Infrastructure Specification

An Edge Cloud Infrastructure consists of a Edge Management Cluster (ONEedge Controller) with the Front-end master nodes and a Cloud Infrastructure with the components needed for offering cloud computing. The Edge Cloud Infrastructure is made of one or several Edge Clusters, which can be located at multiple geographical locations, with the hypervisor nodes and the storage system, all interconnected with multiple networks for internal storage and management node communication, and for private and public guest (VM or container) communication.

Edge Cloud's Management services, including scheduling, monitoring and life-cycle management, run in the cloud front-end and orchestrate the edge clusters. The front-end also provides access to the administration tools, user interfaces and API. Although the requirements may vary depending on the number and size of the clusters and API load, the front-end node only requires 8 GB of main memory and 4 cores. The basic building blocks of the architecture, including the edge clusters, are depicted in the next figure:

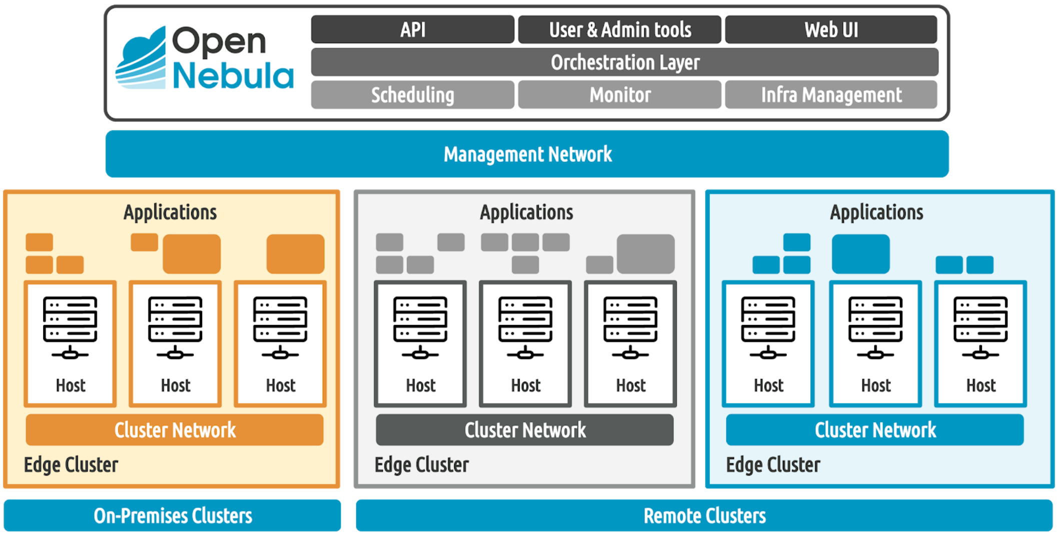

Figure 1.1. Main components of the Edge Cloud Architecture.

An edge cluster is a hyper-converged functional set of managed objects that include storage, network and host resources. An edge cluster provides all the resources needed to run virtualized or containerized applications. The virtualization hosts are responsible for providing applications with execution resources (e.g. CPU, memory, or network access) through a suitable virtualization form, i.e. virtual machines (QEMU/KVM), system containers (LXC) or microVMs (Firecracker/KVM). The actual virtualization technology depends on how the cluster hosts are provisioned and the profile of the workload. Additionally, hosts supply the cluster with storage space needed to run the applications.

The configuration of the hosts of a cluster is homogeneous in terms of the installed software components, OpenNebula administration user, and accessible storage. The front-end nodes need to be connected to the cluster hosts to receive status and monitoring updates as well as to initiate management operations. Cluster nodes are interconnected through one or more private networks, normally used for storage transfers as well as for application communication across virtualization hosts. Finally, access to a public link is required to provide applications with internet connectivity.

The Edge Cloud Architecture is able to provide a lightweight and easy-to-use storage platform for medium-sized clusters consisting of tens of nodes that can run on-premises and on-cloud, and on physical and on virtualized resources. Overall the ONEedge Cloud architecture is able to manage hundreds of these clusters, as they operate autonomously in terms of networking and storage, and handle thousands of virtualized hosts and tens of thousands of virtualized applications.[2]

1.1. Front-end Deployment

The maximum number of virtual instances and servers (virtualization hosts) that can be managed by a single edge cloud front-end instance strongly depends on the performance and scalability of the underlying cloud architecture and platform infrastructure, mainly the storage subsystem.

The capacity specifications and configurations described in this section have been tested for installations running up to 20,000 VMs on 1,250 hosts.[3] The differences in the underlying hardware and performance tuning can result in varying capabilities even between similar configurations. The general recommendation is that no more than 2,500 servers should be managed by a single front-end.

You can scale down your resources to the requirements of your target workload. For example when running the front-end in a cloud resource, you can use t1.small.x86[4] in Packet (Equinix Metal) or t2.medium[5] in AWS if you are planning to deploy medium-scale clouds. The front-end sustains the Image Datastores of the distributed cloud, and it stores the catalog of the apps available in the cloud. The storage area of the front-ends needs to be dimensioned accordingly.

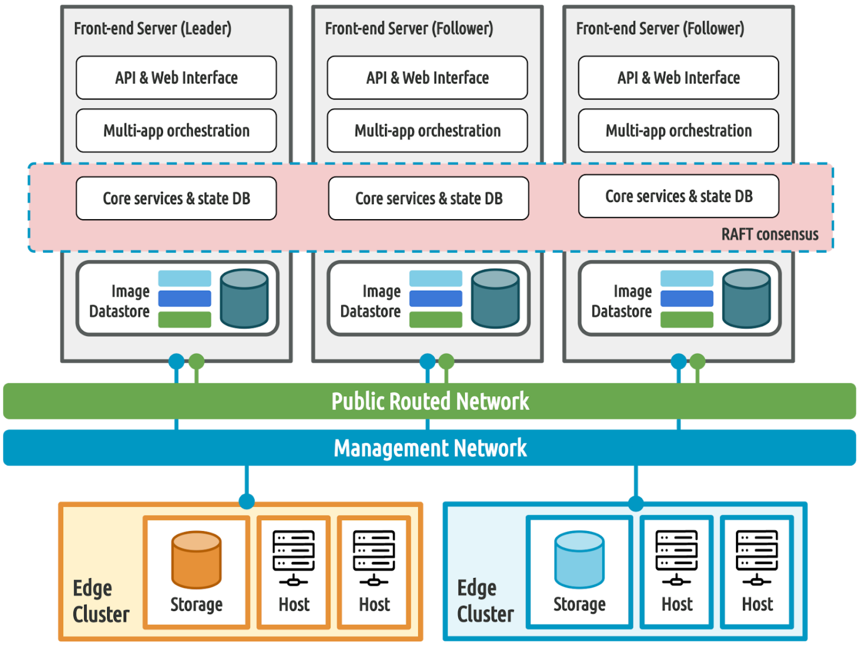

Figure 1.2. Front-ends in a HA configuration and interconnection with edge clusters.

[2] https://docs.opennebula.io/stable/deployment/references/one_scalability.html#front-end-oned-scalability
[3] http://docs.opennebula.io/5.12/deployment/references/one_scalability.html
[4] https://www.packet.com/cloud/servers/t1-small/
[5] https://aws.amazon.com/ec2/instance-types/

ON-PREMISES FRONT-END CONFIGURATION	
Number of Servers	1 (3 servers needed in HA configuration)
Server Specs	CPU: 1 CPU with 2 physical cores @ 2.2 GHz RAM: 16GB Disk: SSD 500 GB (must be scaled to accommodate the Image Datastore) NICs: 2x10Gb/s
Operating System	RHEL/CentOS 8
Networking	VLAN (802.1Q linux bridging) ● 1x10Gb/s VLAN for Management Network ● 1x10Gb/s VLAN for Public Routed Network
Connectivity Requirements (inbound)	Traffic in Public routed network ● 443 Sunstone Web UI ● 2633 oned XMLRPC API ● 5030 OpenNebula Gate ● 2474 Flow ● 29876 VNC connections to instances Traffic in Management network ● 22 SSH ● 4124 Monitor and status messages from clusters

PACKET (EQUINIX METAL) - FRONT-END CONFIGURATION	
Number of Servers	1 (3 servers needed in HA configuration)
Instance Specs	c3.small.x86[6] metal (disk size must be scaled to meet Image Datastore requirements)
Location	Any (3 different AZ recommended for HA configuration)
Operating System	RHEL/CentOS 8
Networking	L2 configuration with multiple VLANs
Connectivity Requirements (inbound)	Traffic in Public Routed Network ● 443 Sunstone Web UI ● 2633 oned XMLRPC API ● 5030 OpenNebula Gate ● 2474 Flow ● 29876 VNC connections to instances Traffic in Management Network ● 22 SSH ● 4124 Monitor and status messages from clusters

[6] https://www.packet.com/cloud/servers/c3-small/

AWS - FRONT-END CONFIGURATION	
Number of Servers	1 (3 servers needed in HA configuration)
Instance Specs	t2.xlarge[7] virtual (disk size must be scaled to meet Image Datastore requirements)
Location	Any (3 different AZ recommended for HA configuration)
Operating System	RHEL/CentOS 8
Networking	VPC with Internet Gateway and subnets
Connectivity Requirements (inbound)	Traffic in Public routed network ● 443 Sunstone Web UI ● 2633 oned XMLRPC API ● 5030 OpenNebula Gate ● 2474 Flow ● 29876 VNC connections to instances Traffic in Management network ● 22 SSH ● 4124 Monitor and status messages from clusters

1.2. Edge Cluster Deployment

The edge cluster capacity specifications and configurations have been defined and tested to hold within each cluster up to 25 nodes, and a single front-end to manage up to 50 clusters, achieving an overall total of 1,250 nodes per cloud instance. The differences in the underlying hardware, performance tuning and distance between clusters can result in varying capabilities even between similar configurations.

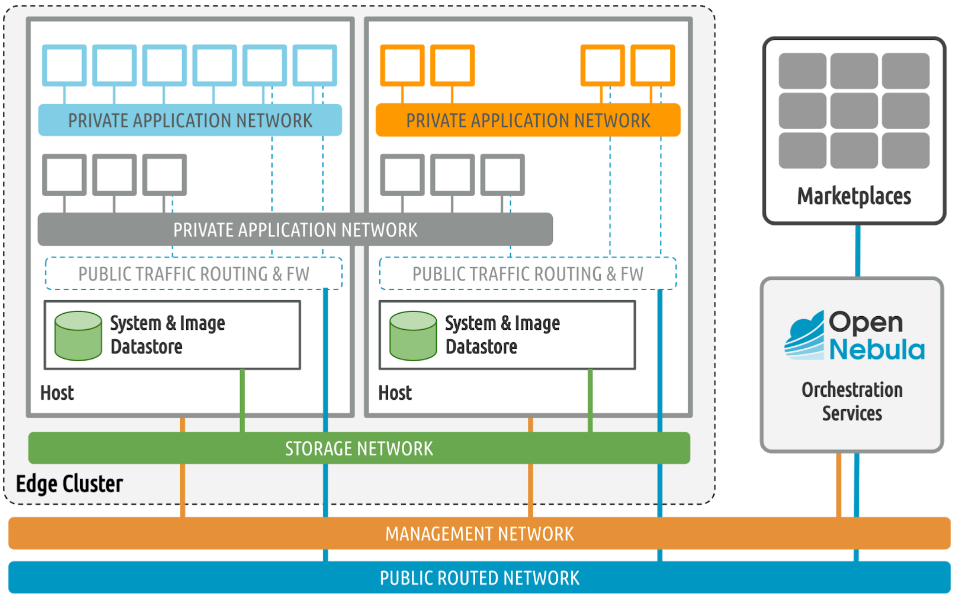

Figure 1.3. Overview of the networks used in the Edge Cloud Architecture.

[7] https://aws.amazon.com/ec2/instance-types/

A key task when defining a cloud infrastructure is to correctly dimension the virtualization nodes according to the expected workload, that is, the characteristics of the VMs that are going to be run in the cloud. The specifications below have been selected so a host can run comfortably 16 medium-size instances (2 virtual CPUs and 16 GB RAM) or up to 500 small-size instances (1 virtual CPU and 512 MB RAM). Alternatively, a host with these specifications can manage up to 1500 microVMs (Firecracker/KVM).

ON-PREMISES - EDGE CLUSTER CONFIGURATION	
Number of Nodes	Up to 25 (it is recommended to configure 2 of them as replica hosts with larger storage capacity for cluster failover)
Server Specs	CPU: 2 CPUs with 32 Physical Cores @ 2.2 GHz RAM: 256GB Disk: 500 GB SSD (2 TB SSD for replica hosts) NICs: 4x10Gb/s
Operating System	RHEL/CentOS 8
Hypervisor	QEMU-KVM / Firecracker-KVM / LXC
Networks	VLAN (802.1Q linux bridging) with VLAN trunk support ● 1x10Gb/s VLAN for Storage Network ● 1x10Gb/s VLAN for Private instances Network ● 1x10Gb/s VLAN for Management Network ● 1x10Gb/s VLAN for Public routed Network
Connectivity Requirements (inbound)	Traffic in Management Network ● 22 SSH for hypervisors ● 5900 and upwards for VNC ports

PACKET (EQUINIX METAL) - EDGE CLUSTER CONFIGURATION	
Number of Nodes	Up to 25 (it is recommended to configure 2 of them as replica hosts with larger storage capacity for cluster failover)
Instance Specs	m3.large.x86[8] metal
Location	Any
Operating System	RHEL/CentOS 8
Networking	L2 configuration with multiple VLANs ● Elastic Public IPs ● VLANs for instances ● Storage VLAN
Connectivity Requirements (inbound)	Traffic in Public Network ● 22 SSH for hypervisors ● 5900 and upwards for VNC ports

[8] https://www.packet.com/cloud/servers/m3-large/

AWS - EDGE CLUSTER CONFIGURATION	
Number of Nodes	Up to 25 (it is recommended to configure 2 of them as replica hosts with larger storage capacity for cluster failover)
Instance Specs	m6gd.metal metal or m6gd.16xlarge virtual[9]
Location	Any
Operating System	RHEL/CentOS 8
Networking	VPC with Internet Gateway and subnets ● Elastic Public IPs ● VPC subnet for storage network ● Internal bridging networking for instances
Connectivity Requirements (inbound)	Traffic in Public Network ● 22 SSH for hypervisors ● 5900 and upwards for VNC ports

1.3. Performance Metrics

The performance improvements in ONEedge cannot be measured in terms of latency or capacity metrics for the edge workloads. ONEedge is an edge computing platform that builds and orchestrates a distributed edge cloud infrastructure to run private edge computing applications using resources from third-party providers. Those providers are the partner enterprises, as discussed in Deliverable D5.1, that will ultimately provide the resources to make up one's edge cloud solution and hence responsible for offering low latencies. Users will be able to select which providers, locations and instances are better suited for their edge applications in terms of cost, capacity, latency, bandwidth, etc.

- In the Centralized Cloud Provider space, we have established partners in **AWS** and **Equinix Metal** (formerly known as Packet), both organizations that provide bare-metal offerings which open the possibility of defining edge platform architectures across various public cloud providers.
- From an Edge Cloud Provider perspective, we have been working very closely with Telefónica, and we are now partnering organizations like **AWS** for their **Wavelength** offering, which provides specific infrastructure deployments that embed AWS compute and storage services right within the data centers at the edge of the growing 5G network of communications service providers (CSP) like Verizon in the US and Vodafone in Germany and the UK.

The performance improvements in ONEedge are measured in terms of performance and scalability of its components. ***Our aim is to keep those metrics while new features are incorporated into the solution.***

Edge Cluster Deployment Performance

We have evaluated the deployment time of an Edge cluster on different providers, zones and instance types. In general the following may influence the overall deployment time:

[9] https://aws.amazon.com/ec2/pricing/on-demand/

- The type of instance, in general bare metal instances are delivered with a higher latency.
- The zone status and load that may influence its responsiveness.
- The location of the ONEedge controller that may also incur in higher control latencies.
- The associated provider elements (VPCs, security groups,...) needed to implement the provision.

The table below shows the average overall deployment time, from 0 to full operational status, in different regions, providers and for different host types (virtualized and bare metal). The ONEedge controller was deployed in Madrid (Spain) for all cases. The provision consists of 3 hosts, a public network with 2 pre-allocated IPs and a private virtual network.

	AWS virtual	AWS metal	Equinix
Europe Zone	325s	562s	416s
US Zone	612s	825s	658s

Table 1.1. Average total deployment times using virtual/metal instances on different providers and zones

The specific zones used in this experiment are described in the next table:

	AWS	Equinix
Europe Zone	eu-west-1 (Ireland)	ams1 (Amsterdam)
US Zone	us-west-1 (N. California)	sjc1 (Sunnyvale)

Table 1.2. European and US zones used during the deployment time experiments.

Application Deployment Performance

Application images are based on files with qcow2 format to reduce file transfer and instantiation times with minimal overhead. The use of qcow2 files to back disk images also eases backup solutions, reduces image transfer times, and implements advanced features like snapshotting in an efficient way. The expected deployment times depend on the interconnection links between tiers and disk sizes. As a reference, for a metal remote cluster (Tier 3) in Frankfurt, connected using public internet links to a Tier 2 image datastore placed in Amsterdam, we have obtained averages deployment times of 5s (hot caches) and 35s (first-time, empty cache) for applications using qcow2 disks of 0.5 GiB. Average deployment times for mentioned clusters with cached and without cached images are described in the next figure. When the clusters are deployed on-premises using 1Gb/s links the deployment time is always 5s.

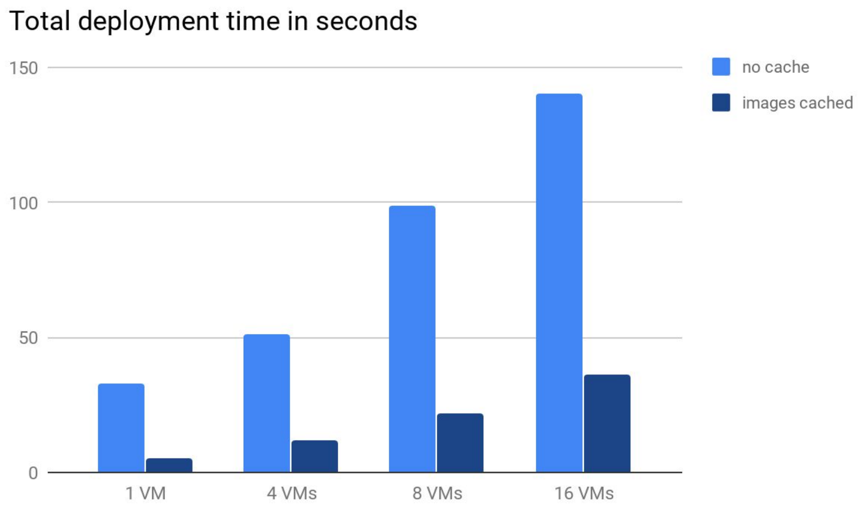

Figure 1.4. Comparison of deployment times with/without cached images.

Application I/O Performance

The applications run from the direct attached storage of the hosts, to maximize the available I/O performance delivered to the applications. The I/O performance is close to that of the native host and it is only impacted by the virtualization layer. In order to provide an estimation of this overhead we have run the Flexible I/O tester on both the host and a VM. The measures have been taken on a t1.small metal instance on Equinix and a VM running on the very same server. The table below shows a summary of the characteristics of both machines.

Physical Server	Virtual Machine
1 x Intel Atom C2550 @ 2.4Ghz 8GB RAM 1 x 80GB SSD DISK NIC 2 x 1Gbps Bonded Ports	KVM kernel 4.18.0-147.8.1.el8_1.x86_64 Libvirt-6.0.0 Qemu (cache=none, io=native)

Table 1.3. Main characteristics of the servers used to run the benchmarks.

The next table shows the average bandwidth obtained while running the benchmark for read and write operations, both sequential and random. The performance is similar between the physical host and VM, with differences less than a 5%. Note that some latency is hidden by the virtual page cache.

		Host	Virtual Machine
Sequential	Read	179 MiB/s	171 MiB/s
	White	168 MiB/s	181 MiB/s
Random	Read	84.8 MiB/s	90.2 MiB/s
	White	90.3 MiB/s	94.8 MiB/s

Table 1.4. Average IO bandwidth for read/write operations on the physical host and virtual machine.

Application Snapshot Performance

In order to improve the availability of the edge cluster blocks, live migration is supported within the same cluster. Application snapshots are also kept within the edge cluster (Tier 3) to enable fast recovery from the last application checkpoint.

The recovery operation can potentially impact on two different areas:

- I/O noise generated by the snapshot operation that could reduce the I/O of neighbour applications. In our case this cost is negligible as it is based on the QEMU Redirect-on-Write feature.
- Network bandwidth to move the recovery snapshots to the cluster replica server. In this case we use a delta-transfer algorithm to reduce the information transferred to the server. However this time will increase as the contents of the disk diverges from the original.

Another important aspect to consider is the VM recovery time. Compared to recovering a VM without any snapshot the time is similar, as the base images are already located on the cluster replica (Tier 3) and the only additional overhead is the transfer of the disk snapshot, already available in the edge cluster as well.

2. Software Requirements Verification

This section includes a detailed list of the extensions implemented to verify the functionality of the software developed in ONEedge during the Second Innovation Cycle (M10-M16). Tests and extensions of the verification framework are detailed for each component and grouped for each software requirement implemented during the cycle. For each requirement we include a summary of the extensions performed in the testing and certification infrastructure. Also, for each requirement, we list the verification scenarios that have been addressed and a description of the functionality tested to fulfill the proposed scenarios.

2.1 Edge Instance Manager (CPNT1)

SR1.1. Simple Product Deployment
Status: IN PROGRESS
Description: Testing has been extended to cover the new type of containerized deployment leveraging various run-time and host operating systems. Verification focuses on the integration testing of the deployment.
Verification Scenarios: • [VS1.1.1] Creates a different type of containerized front-end deployments (single or multi-container) with different container run-times (Docker, Podman) on various base operating systems (CentOS, Ubuntu, Debian). • [VS1.1.2] Tests the deployment by adding non-containerized KVM nodes and running integration tests used by natively installed KVM environments with SSH datastore.

SR1.2. Automatic Product Upgrade
Status: IN PROGRESS
Description: Testing of containerized front-end deployment, where bootstrap and service scripts inside the container refresh the persistent data to the newer version of product.
Verification Scenarios: • [VS1.2.1] Functionality verified as part of [VS1.1.1] and [VS1.1.2].

SR1.3. Instance Management
Status: IN PROGRESS

Description: Tests have been extended to create a deployment with non-default configuration parameters to leverage the automatic reconfiguration.
Verification Scenarios: • [VS1.3.1] Tests configure EdgeStack containers based on custom configuration parameters. Deployment is automatically reconfigured during the bootstrap and functionality verified by the integration tests. Rest is verified as part of [VS1.1.1] and [VS1.1.2].

2.2 Edge Workload Orchestration and Management (CPNT2)

SR2.2. Specialized Cache Datastore
Status: DONE
Description: Testing of replica based datastore with recovery snapshot capabilities
Verification Scenarios: • [VS2.2.1] Tests that deploy a VM using a replica datastore, cache contents are verified after deployment • [VS2.2.2] Test to deploy a VM from a *"hot"* cache with and without a recovery disk snapshot.

SR2.4. Virtual Machine Management Operations: Backups
Status: IN PROGRESS
Description: Tests that define several backup policies and verify the recovery process.
Verification Scenarios: • [VS2.4.1] Tests that define several backup frequencies and verify that backups are created at the specified points in time • [VS2.4.2] Tests to verify backup integrity in the designated storage area • [VS2.4.3] Test the recovery procedure of a VM from an existing backup

SR2.8 Complete Service Flows
Status: DONE

Description: Additional tests to check the new event based life cycle manager
Verification Scenarios: ● [VS2.7.3] Tests to verify the scalability of the Flow service when a growing number of services and VMs

SR2.9. Web UI extensions
Status: DONE
Description: Sunstone tests have been extended to cover the new functionality exposed by the interface. All sunstone tests are end to end, meaning that full stack (including the infrastructure drivers) are exercised.
Verification Scenarios ● [VS5.1.1] [VS5.1.2] Tests cover that a user can use Sunstone to import a full OneFlow service template as well as a full VM template into a local datastore. ● [VS2.4.1] [VS2.4.2] Test the recovery procedure of a VM from an existing backup using Sunstone. Tests cover that a user can successfully add backup frequency and marketplace target for a given VM from Sunstone.

SR2.10. LXC virtualization drivers for OpenNebula
Status: DONE
Description: Testing new LXC drivers.
Verification Scenarios: ● Tests that deploy VMs and perform basic operations using LXC based hosts. ● Tests that exercise the different mapper modules: raw, qcow2 and rbd ● Tests that uses different network connectivity drivers for LXC containers

2.3 Edge Provider Selection (CPNT3)

SR3.1. Edge Provider Catalog Service
Status: IN PROGRESS

Description: New set of tests were developed using the existing ruby RSpec framework to exercise the provider management through the oneprovision CLI command. These are end to end tests that create aws and packet providers with oneprovision, using real cloud provider credentials, and exercise the management logic. It calls the Fireedge server directly, without exercising the Javascript client.

Verification Scenarios:

- [VS3.1.1] Test for provider listing, both instances and templates. Test for provider management: creation, update and delete. They exercise both supported to date providers, aws and packet.

SR3.4 Driver Maintenance Process

Status: IN PROGRESS

Description: The goal is that any new third party providers need to have a process to develop, test, contribute and certify its drivers with new versions of ONEedge. The set of tests that are used to exercise the whole provision automatic procedure in ONEedge is the basis of the driver maintenance process.

Verification Scenarios:

- [VS3.4.3] The current provider implementation of aws and packet passes the driver maintenance process tests.

SR3.5 Edge Catalog Web Interface

Status: DONE

Description: New set of tests were developed for Fireedge, and in particular for the OneProvision GUI. In particular, this tests exercise only the Provider tab of the GUI, using Chrome as the chosen explorer exercising the full stack from the Javascript client.

Verification Scenarios:

- [VS3.5.1] Test for provider listing using the card interface of OneProvision GUI. Test for provider management: creation, update and delete. Creation of a provider includes the exercise of the wizard-like stepper implemented in the GUI

2.4 Edge Infrastructure Provision and Deployment (CPNT4)

SR4.1 Reliable Edge Resource Provision

Status: DONE

Description: Test the retry loop of the provision engine
Verification Scenarios: • [VS4.1.1.] Test different retry operations for each provision step. Test that each step is reported. • [VS4.1.2.] Tests that verify that edge provider resources are successfully deallocated after a provision removal or failure

SR4.2 Usability, Functionality and Scalability of Provision
Status: IN PROGRESS
Description: Description: Extend networking tests with public and private networks
Verification Scenarios: • [VS4.2.1.] Extend cycle 1 verification tests to cover AWS and Equinix/Packet providers

SR4.3 Provision Template for Reference Architectures
Status: DONE
Description: Tests to deploy the reference architecture on the set of supported providers
Verification Scenarios: • [VS4.3.1] Test to deploy the reference architecture in AWS and Equinix/Packet. Basic functionality operations are performed by the tests to check that the provision is operational

SR4.5. Drivers for Host Provision
Status: IN PROGRESS
Description: Tests to deploy advanced provisions consisting of different cloud resources.
Verification Scenarios: • [VS4.5.1.] Test to deploy provisions that include support resources including: security group rules, routing information, and storage blocks.

SR4.6. Drivers for IP Address Management
Status: DONE
Description: Developed as part of SR4.7

SR4.7. Drivers for Network Drivers and Helpers
Status: DONE
Description: Tests that provision different network topologies and verify the connectivity of applications.
Verification Scenarios: • [VS4.6.1.] Tests that allocate elastic IPs in different providers and successfully release them. • [VS4.7.1.] Test that attach a previously allocated IP to a VM in a provision. The test checks that the IP is assigned to the right instance and traffic is forwarded to the right VM. • [VS4.7.2.] Test public connectivity of a VM in a provision. The hypervisor cannot be accessed by the VM. • [VS4.7.3.] Test that allocates a private network on a provision. Several VMs are attached to this network and connectivity is verified by generating application network traffic.

SR4.8. GUI for Edge Resource Provision
Status: DONE
Description: New set of tests were developed for Fireedge, and in particular for the OneProvision GUI. These tests are based on Cypress testing framework, and are end to end tests, exercising the whole stack. In particular, these tests are focused on testing provision management.
Verification Scenarios: • [VS4.8.1] Tests to ensure a user can create, update and new provisions (ie OpenNebula clusters on remote locations) from a set of templates (currently aws and packet) • [VS4.8.2] Tests to ensure a user can control remote hosts state, including troubleshooting through SSH sessions the hosts.

2.5 Edge Apps Marketplace (CPNT5)

SR5.1. Edge Applications and Services in Marketplace
Status: DONE
Description: These set of tests add to the existing OpenNebula RSpec test suite. As other tests of the suite they are end to end tests that use the public OpenNebula marketplace to import existing resources into a test environment.
Verification Scenarios: • [VS5.1.1] Tests cover that a user can use Sunstone to import a full VM Template with multiple disks into a local datastore. • [VS5.1.2] Tests cover that a user can use Sunstone to import a full OneFlow service template into a local datastore.

SR5.2 Built-in Management of Application Containers Engine
Status: IN PROGRESS
Description: A new test for the updated Kubernetes appliance has been developed using the existing RSpec and Sunstone Selenium test frameworks. In particular, this new test exercises the multi-node deployment of a Kubernetes cluster. Additionally a separate test for K3s was implemented leveraring the Dockerfile functionality.
Verification Scenarios: • [VS5.2.1] Test that the Kubernetes appliance, as defined in the OpenNebula public marketplace, can be imported as a OneFlow Service Template automatically • [VS5.2.2] Test Kubernetes OneFlow Service Template can be instantiated and that elasticity rules are correctly applied. • [VS5.2.3] Test the whole end to end procedure of defining a k3s cluster from a Dockerfile.

SR5.3 Integration with Application Containers Marketplace
Status: DONE
Description: The new functionality is tested as an extension of the tests developed in cycle one. In particular this new test goes through the process of defining a local dockerfile and importing it into opennebula

Verification Scenarios:

- [VS5.3.1] Test that a local dockerfile can be imported into OpenNebula using the dockerfile://localpath syntax. Test that the resulting container in OpenNebula can be correctly instantiated

SR5.5. Edge Market GUI Developments
Status: IN PROGRESS
Description: A new set of tests for FireedgeFlow are being developed. Currently a subset of the functionality of the Technology Preview can be defined. This leverages the Cypress testing framework.
Verification Scenarios: • [VS5.5.1] Test that a user can define a complex OneFlow service from the GUI, mixing Virtual Machines and containers imported from DockerHub • [VS5.5.2] Test that a user can instantiate and manage OneFlow services in a particular OpenNebula cluster, which can be in a remote (edge) location.