The Role of Scripting and Automation in Modern Cloud Operations

University of Essex Online
Tobias Zeier, 20 August 2025

Automation through scripting is a foundational practice in modern cloud computing, significantly enhancing operational efficiency, consistency, and scalability. This approach, exemplified by the provided *provision.sh* script, allows for the repeatable and reliable deployment of cloud infrastructure, mitigating the risks associated with manual processes. Scripting transforms complex, multi-step procedures into a single, automated workflow, thereby reducing human error and freeing up personnel to focus on more strategic initiatives. This aligns with contemporary Site Reliability Engineering (SRE) principles, which advocate for the automation of routine tasks to improve system reliability and reduce downtime (Dave, 2023).

The Role of Infrastructure as Code

The *provision.sh* script serves as a practical example of Infrastructure as Code (IaC), a paradigm where infrastructure is managed and provisioned using code rather than through manual configuration (Nuti, 2025). This method ensures that the infrastructure environment is version-controlled, testable, and reusable. For instance, the script's core commands are designed to be idempotent, meaning they can be run multiple times without causing unintended side effects. The script checks for the existence of networks, security groups, and images before creating them, ensuring a consistent and predictable state with every execution. This addresses a key challenge in large-scale cloud environments, where maintaining operational consistency is paramount to preventing misconfigurations and ensuring system stability (Koneru, 2025).

Efficiency, Scalability, and Consistency

The primary benefit of cloud automation is the dramatic increase in operational speed and efficiency. Manually configuring cloud resources is a time-consuming and error-prone process. By automating these tasks, the time to deploy new environments or scale existing ones is reduced from hours to mere minutes (Madamanchi, 2025). The *provision.sh* script showcases this by automating the entire lifecycle of a virtual instance, from network setup to floating IP assignment. The script’s manual polling loop to wait for the instance to become 'active' is a robust automation pattern that eliminates manual checks and retries.

This automation also facilitates unprecedented scalability. When an organisation's demands increase, scripts can be readily adapted to provision hundreds or thousands of instances on demand. The logic in the *provision.sh* script to check for and reuse an unassigned floating IP address highlights another key aspect of smart automation: resource optimisation. By automating the management of these resources, an organisation can avoid unnecessary costs associated with creating new floating IPs when existing ones are available. This demonstrates how scripting can lead to significant cost savings and more sustainable cloud usage (Nuti, 2025). In essence, the move towards scripting in cloud operations is a strategic shift from manual, reactive management to a proactive, codified approach. The provided script, although simple in its structure, embodies the core tenets of modern DevOps and cloud engineering: it prioritises automation to achieve greater speed, maintain a consistent state, and enable scalable, cost-effective infrastructure management (Dave, 2023).

Word count: 459

References

Madamanchi, S.R. (2025) 'Modern approaches to Unix automation: Shell scripting, configuration management, and security,' *International Journal for Research in Applied Science and Engineering Technology*, 13(6), pp. 3190–3201. Available at: https://doi.org/10.22214/ijraset.2025.72773.

Dave, D.M. (2023) 'Impact of site reliability engineering on manufacturing operations: Improving efficiency and reducing downtime,' *International Journal of Scientific and Research Publications*, 13(11), pp. 136-138. Available at: https://doi.org/10.29322/ijsrp.13.11.2023.p14312

Koneru, N.M.K. (2025) 'Infrastructure as Code (IAC) for Enterprise Applications: A Comparative Study of Terraform and CloudFormation,' *American Journal of Technology*, 4(1), pp. 1-29. Available at: https://doi.org/10.58425/ajt.v4i1.351

Nuti, S. (2025) ‘Optimizing Cloud Service Delivery with Infrastructure as Code (IaC) and Platform Engineering’, *Journal of Recent Trends in Computer Science and Engineering*, 13(4), pp. 1-13. Available at: https://doi.org/10.70589/JRTCSE.2025.13.4.1

Appendix

The screenshots provided document the successful execution of the *provision.sh* bash script. A comprehensive log file of the exact output, as well as the bash script itself, will be submitted alongside this document for review.

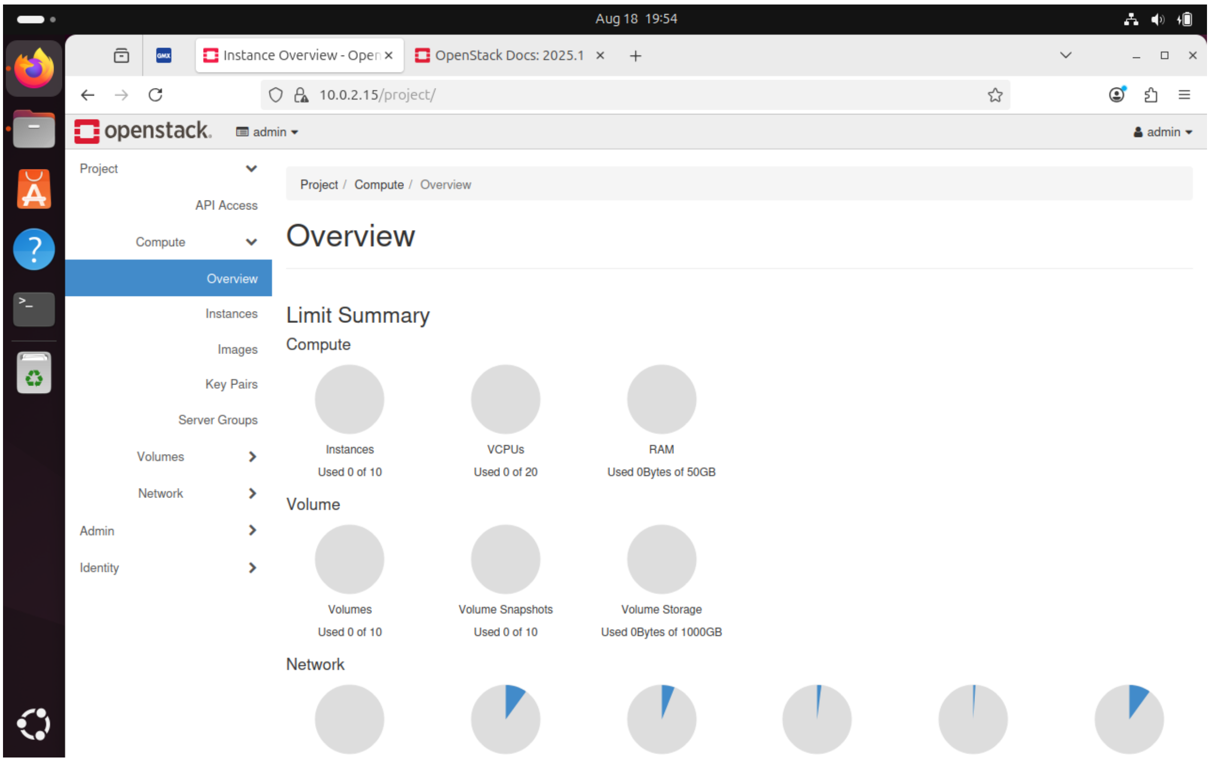
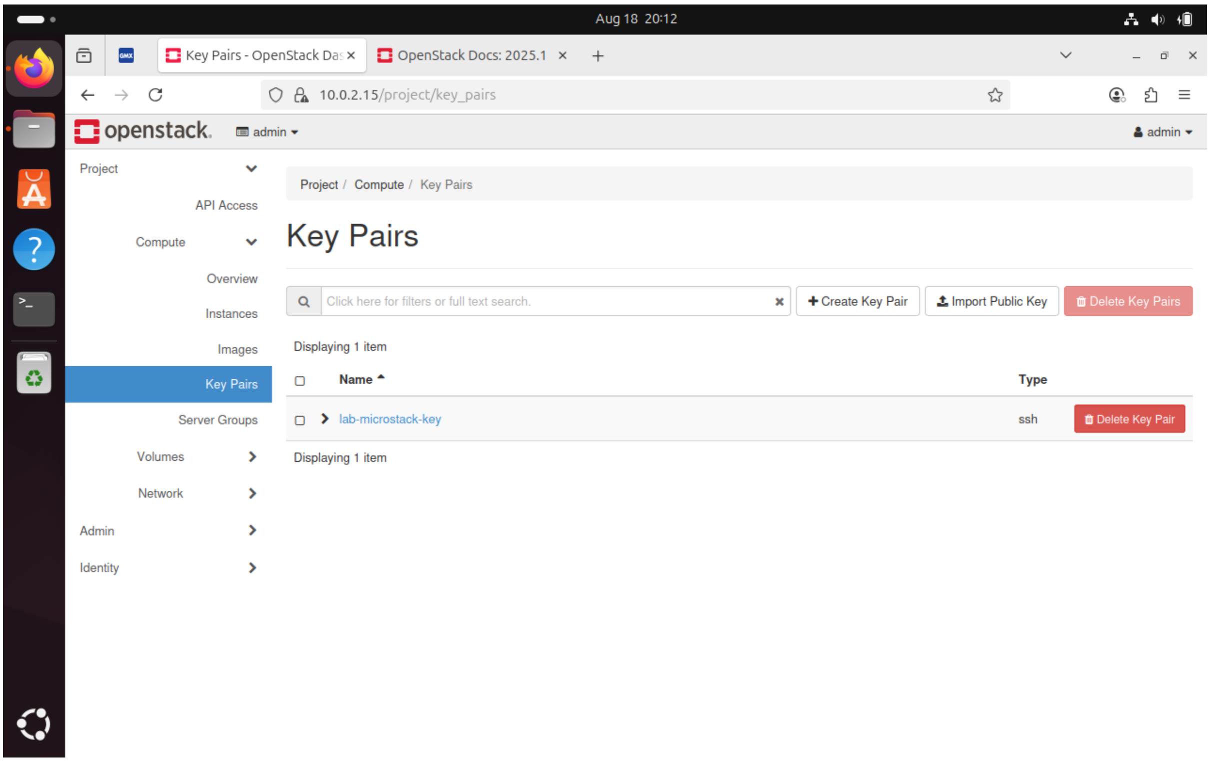


Figure 1: An Empty MicroStack Environment



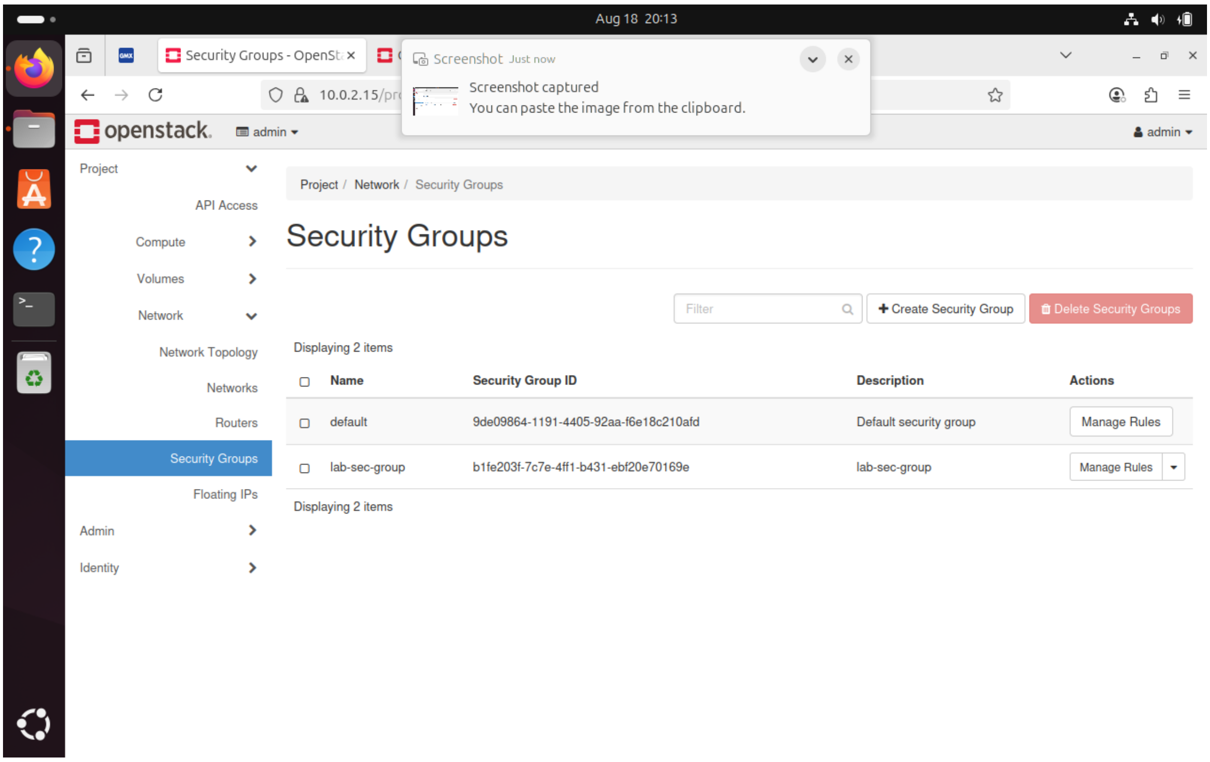
Key Pairs

Displaying 1 item

☐	Name	Type	
☐	lab-microstack-key	ssh	Delete Key Pair

Displaying 1 item

Figure 2: Automatically created Key Pairs



Screenshot Just now

Screenshot captured
You can paste the image from the clipboard.

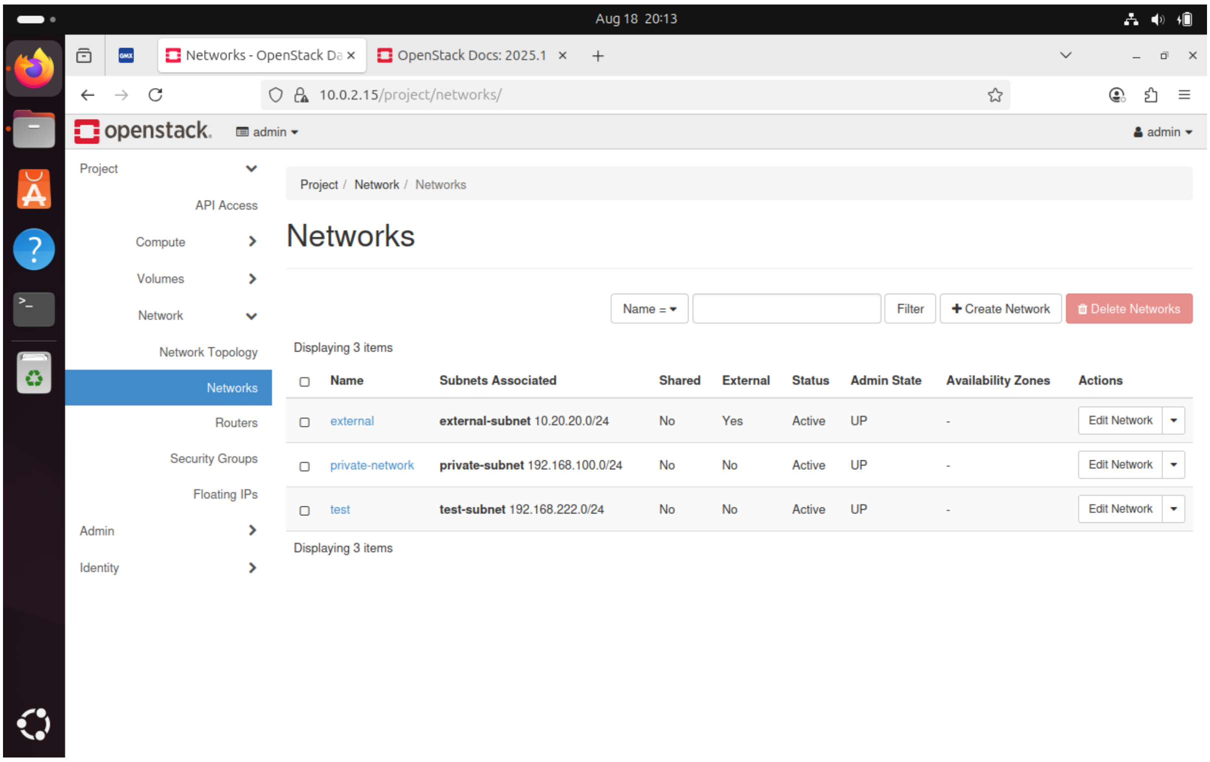
Security Groups

Displaying 2 items

☐	Name	Security Group ID	Description	Actions
☐	default	9de09864-1191-4405-92aa-f6e18c210afd	Default security group	Manage Rules
☐	lab-sec-group	b1fe203f-7c7e-4ff1-b431-ebf20e70169e	lab-sec-group	Manage Rules

Displaying 2 items

Figure 3: Automatically created Security Groups

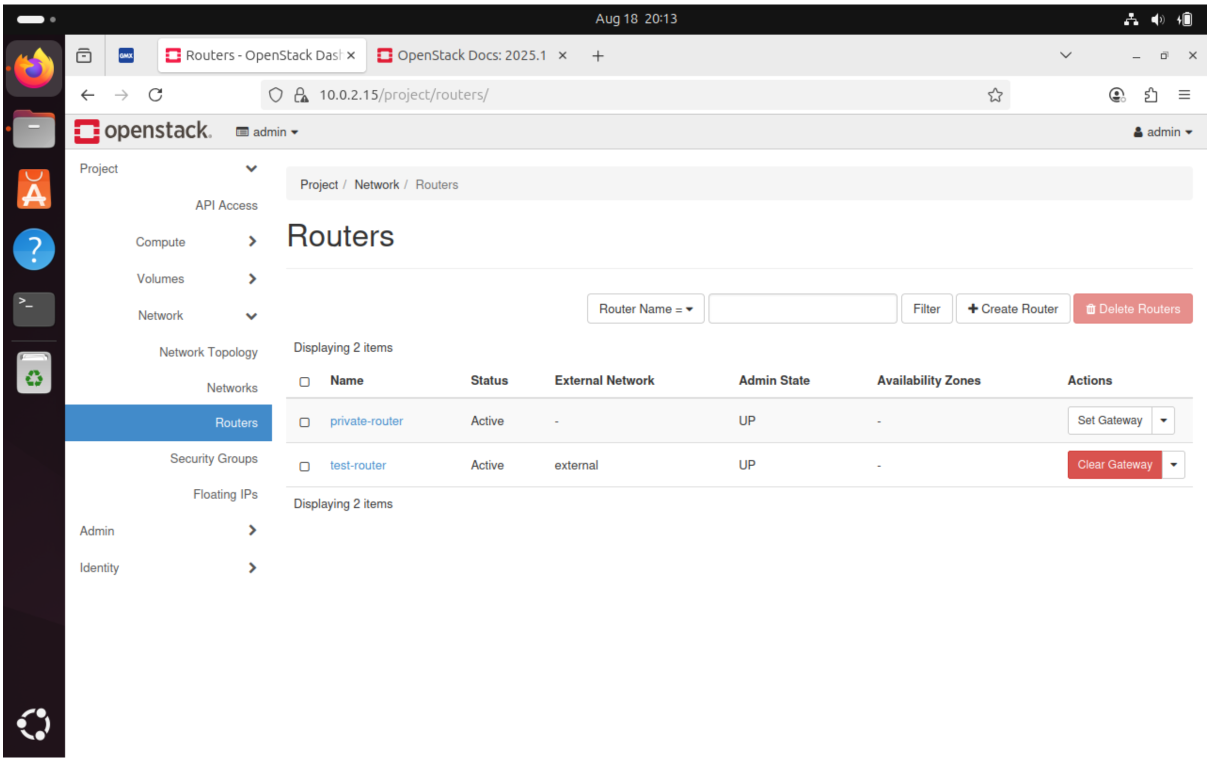


Displaying 3 items

□	Name	Subnets Associated	Shared	External	Status	Admin State	Availability Zones	Actions
□	external	**external-subnet** 10.20.20.0/24	No	Yes	Active	UP	-	Edit Network
□	private-network	**private-subnet** 192.168.100.0/24	No	No	Active	UP	-	Edit Network
□	test	**test-subnet** 192.168.222.0/24	No	No	Active	UP	-	Edit Network

Displaying 3 items

Figure 4: Automatically created Networks

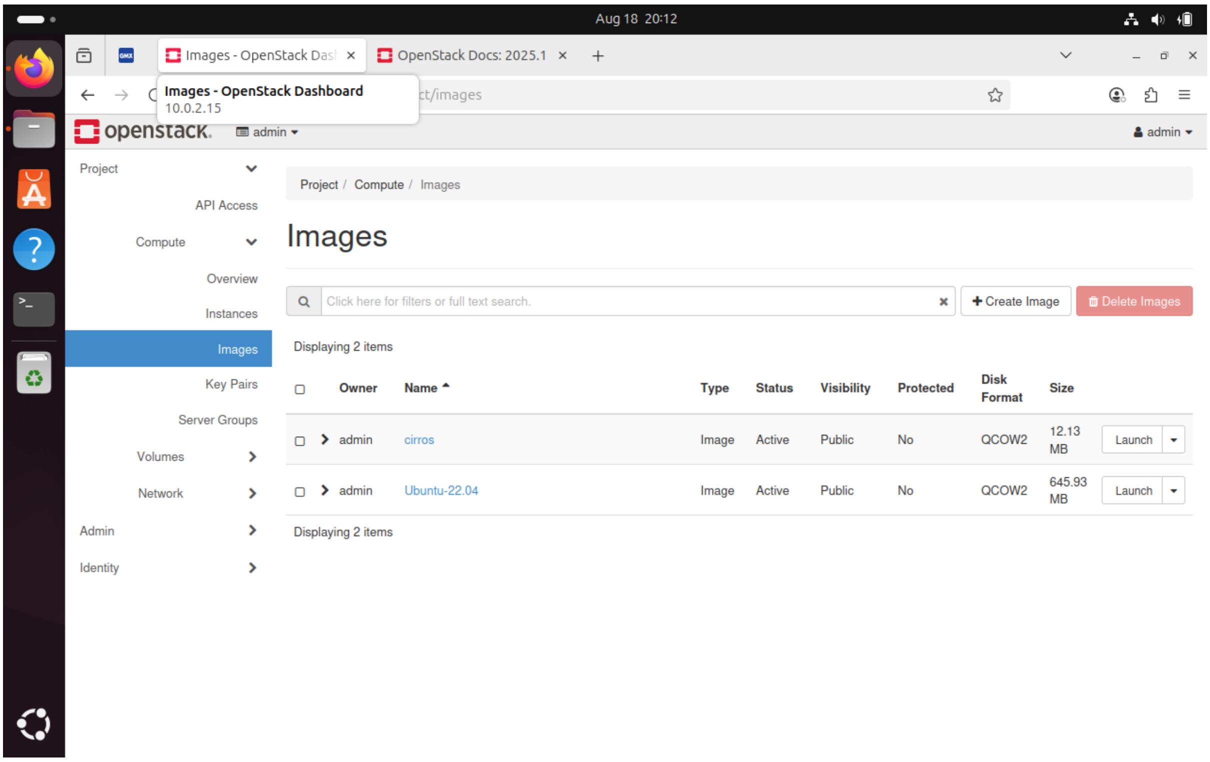


Displaying 2 items

□	Name	Status	External Network	Admin State	Availability Zones	Actions
□	private-router	Active	-	UP	-	Set Gateway
□	test-router	Active	external	UP	-	Clear Gateway

Displaying 2 items

Figure 5: Automatically created Routers



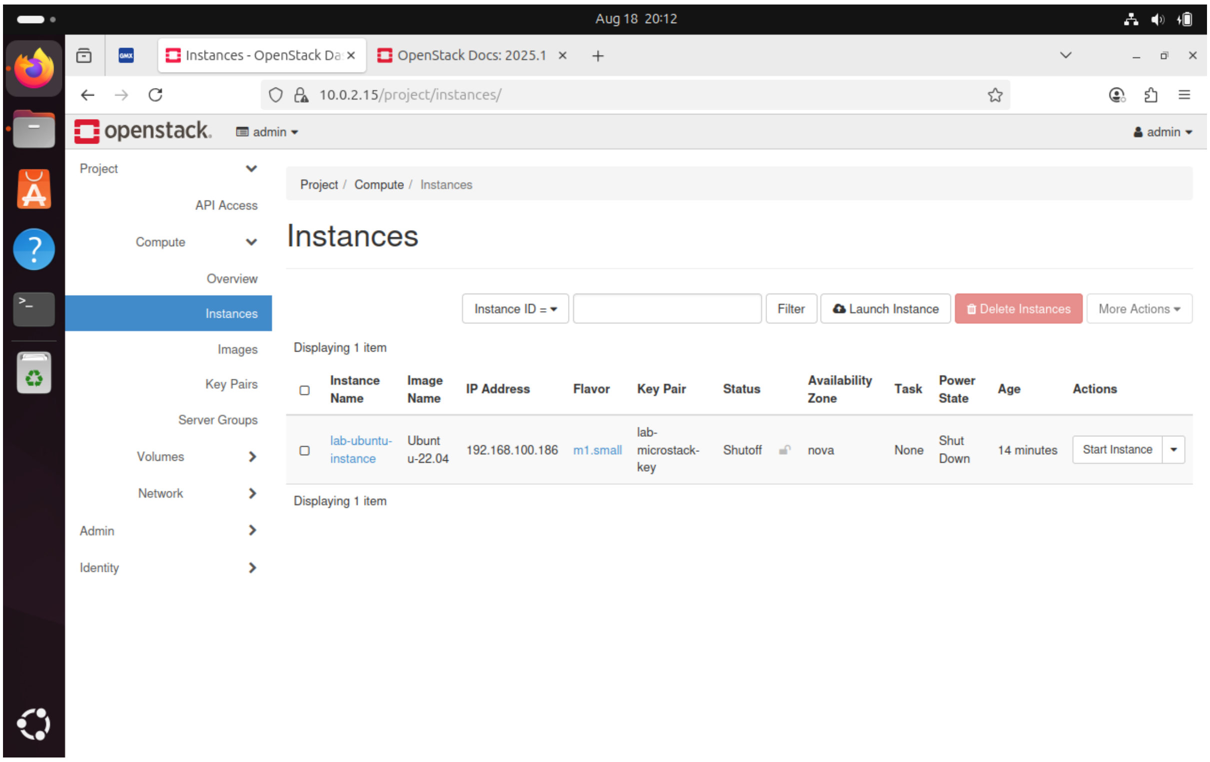
Images

Displaying 2 items

□		Owner	Name	Type	Status	Visibility	Protected	Disk Format	Size	
□	>	admin	cirros	Image	Active	Public	No	QCOW2	12.13 MB	Launch
□	>	admin	Ubuntu-22.04	Image	Active	Public	No	QCOW2	645.93 MB	Launch

Displaying 2 items

Figure 6: Automatically uploaded Images



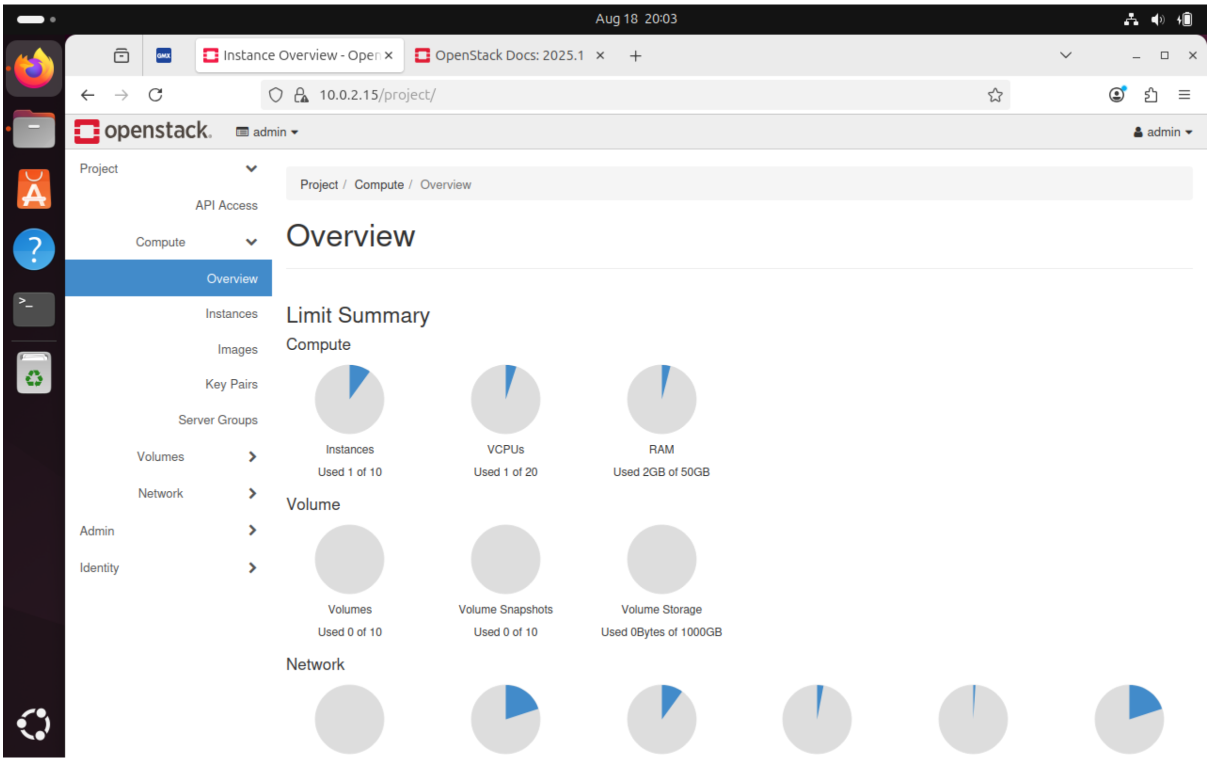
Instances

Displaying 1 item

□	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
□	lab-ubuntu-instance	Ubuntu-22.04	192.168.100.186	m1.small	lab-microstack-key	Shutoff	nova	None	Shut Down	14 minutes	Start Instance

Displaying 1 item

Figure 7: Automatically created Instance



Project / Compute / Overview

Overview

Limit Summary

Compute

Instances
Used 1 of 10

VCPUs
Used 1 of 20

RAM
Used 2GB of 50GB

Volume

Volumes
Used 0 of 10

Volume Snapshots
Used 0 of 10

Volume Storage
Used 0Bytes of 1000GB

Network

Figure 8: Overview after successful execution of Bash Script