Issues we came across with Private Cluster Creation

| **Topic** | **Error Message** | **Mitigation Plan** |
| --- | --- | --- |
| Encryption Error | 2023-07-05T08:36:30.5900122Z [31m│[0m [0mResource Name: "Eslzcmcaks-poc-001"): managedclusters.ManagedClustersClient#CreateOrUpdate: Failure sending request: StatusCode=0 -- Original Error: Code="SubscriptionNotEnabledEncryptionAtHost" Message="Subscription does not enable EncryptionAtHost."[0m | Encryption at Host has been set to true for VMSS as in ES@Uniper the deployments of VMSS is not allowed without encryption. |
| Image Cleaner Error | Error: unrecognized arguments: --enable-image-cleaner | We had to register the below service provider in the sandbox subscription as a pre-requisite.  az feature register --namespace "Microsoft.ContainerService" --name "EnableImageCleanerPreview" |
| Infra Node RG creation Error | 2023-07-05T09:19:53.2210494Z [31m│[0m [0mResource Name: "Eslzcmcaks-poc-001"): Code="RequestDisallowedByPolicy" Message="Create or update resource group MC\_rg-weu-poc-eslz-01\_Eslzcmcaks-poc-001\_westeurope failed" Details= | Default MC\_ RG is not accepted in ES@Uniper due to the policies, hence for the Infra RG as well we are following ES@Uniper naming standards. |
| Vmss creation failure |  | Self-hosted Agent Pool was unable to connect to the storage account in the sandbox subscription where the terraform state file is saved, added the aks Vnet to the Storage account network settings for the connectivity to be established. |
| Vmss Extension error/ outbouund connevtivity | 2023-07-07T16:18:39.2445018Z [31m│[0m [0mResource Group Name: "rg-weu-poc-eslz-03"  2023-07-07T16:18:39.2459858Z [31m│[0m [0mResource Name: "Eslzcmcaks-poc-022"): Code="CreateVMSSAgentPoolFailed" Message="Unable to establish outbound connection from agents, please see <https://learn.microsoft.com/en-us/troubleshoot/azure/azure-kubernetes/error-code-outboundconnfailvmextensionerror> and <https://aka.ms/aks-required-ports-and-addresses> for more information. Details: VMSSAgentPoolReconciler retry failed: Category: ClientError; Code: VMExtensionProvisioningError; SubCode: OutboundConnFailVMExtensionError; Message: Unable to establish outbound connection from agents, please see <https://learn.microsoft.com/en-us/troubleshoot/azure/azure-kubernetes/error-code-outboundconnfailvmextensionerror> and <https://aka.ms/aks-required-ports-and-addresses> for more information. Details: instance 2 has extension error details : {vmssCSE error messages : {vmssCSE exit status=50, output=\_DV=cuda-525.85.12\n++ | With Userdefined route settings as outbound type, the traffic is expected to go via NVA, UDR attached to the subnets with NVA as next hop and once the traffic hits NVA it fails to hit MCR to download the vMSS extensions. Discussed with Network team and shared the list of FQDNs and have raised a FCR 9241 |

List of Destination FQDNs to be whitelisted with AKS Subnet as Source...

|  |  |
| --- | --- |
| **\*.hcp.westeurope.azmk8s.io** | **HTTPS:443** |
| **mcr.microsoft.com** | **HTTPS:443** |
| **\*.data.mcr.microsoft.com** | **HTTPS:443** |
| **management.azure.com** | **HTTPS:443** |
| **login.microsoftonline.com** | **HTTPS:443** |
| **packages.microsoft.com** | **HTTPS:443** |
| **acs-mirror.azureedge.net** | **HTTPS:443** |

A computer screen with text

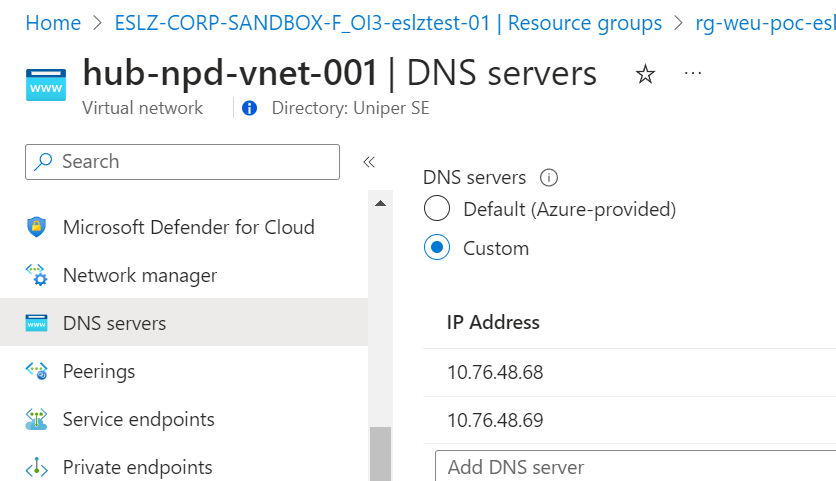
Description automatically generated

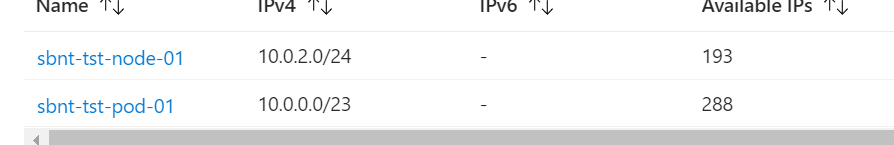
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated





A white box with black text

Description automatically generated

Kubernetes is a powerful open-source platform for managing containerized applications and services. It has become increasingly important in the world of cloud computing and DevOps for several reasons:

1. **Scalability**: Kubernetes allows you to scale your applications seamlessly. Whether you need to handle sudden spikes in traffic or gradually increase capacity, Kubernetes can automatically adjust the number of container instances.
2. **Reliability**: It provides high availability for applications by automatically restarting failed containers, rescheduling them on different nodes, and performing health checks.
3. **Portability**: Kubernetes works with a variety of container runtimes and cloud providers, allowing you to deploy your applications in a consistent manner across different environments, from on-premises data centers to public clouds.
4. **Efficiency**: Kubernetes optimizes resource utilization by packing containers efficiently on nodes. It can help reduce infrastructure costs by ensuring that resources are used effectively.
5. **Automation**: With Kubernetes, you can automate many aspects of application deployment and management, including rollouts, rollbacks, and monitoring. This helps teams to focus on coding and innovation rather than manual operations.
6. **Extensibility**: Kubernetes is highly extensible, with a rich ecosystem of plugins and extensions that can be used to enhance its functionality. This allows you to customize it to fit your specific needs.
7. **Container Orchestration**: Kubernetes automates the deployment, management, and scaling of containerized applications. This orchestration ensures that your application components are running in the correct places, can communicate with each other, and are always in the desired state.
8. **Efficiency**: Kubernetes can distribute workloads across multiple nodes in a cluster, maximizing resource usage and minimizing costs. This means you can run applications more efficiently and scale resources dynamically based on demand.
9. **Consistency**: Kubernetes ensures that your applications run the same way across different environments, from development to production. This consistency reduces the "it works on my machine" problem, making it easier to develop, test, and deploy applications.
10. **Self-Healing**: Kubernetes provides self-healing capabilities by automatically replacing or restarting containers that fail or become unresponsive. This reduces downtime and ensures high availability.
11. **Automation**: Many tasks that would otherwise require manual intervention, such as deployment, scaling, and monitoring, can be automated using Kubernetes. This automation increases operational efficiency and reduces the risk of human error.
12. **Extensibility**: Kubernetes has a rich ecosystem of plugins and integrations, allowing you to extend its capabilities and customize it to fit your specific requirements. This flexibility makes it a versatile platform for a wide range of use cases.
13. **Community and Ecosystem**: Kubernetes has a large and active community of developers and users. This means you have access to a wealth of resources, support, and tools to help you get the most out of the platform.

By leveraging these benefits, Kubernetes helps organizations build and manage resilient, scalable, and efficient applications. Are you thinking about integrating Kubernetes into your projects?

A screenshot of a computer

Description automatically generated

**Key Differences**

* **Architecture**: VMs virtualize hardware to run multiple OS instances, whereas containers virtualize the OS to run multiple applications.
* **Overhead**: VMs have higher resource overhead due to running full OS instances, while containers are more lightweight and resource-efficient.
* **Performance**: Containers generally offer better performance and faster start times due to their lightweight nature.
* **Usage**: VMs are better suited for applications requiring strong isolation and security, whereas containers excel in environments where speed, efficiency, and portability are critical.

In summary, VMs provide a higher level of isolation and security by running full OS instances, but come with more resource overhead. Containers are lightweight and efficient, offering faster performance and easier portability, but with less isolation since they share the host OS kernel.