Nacha Backup Issue Troubleshoot :-

Issue 1 -Created cluster with set authorized IP range. Error While adding a New nodepool.

Error- Failed to add node pool 'phixiusprod' to Kubernetes service 'Nacha-backup'. Error: API server authorized IP ranges requires public IP prefix to enable node public IP.

Solution-

t looks like you're encountering an issue related to the API server authorized IP ranges in Azure Kubernetes Service (AKS). This error occurs because the node pools using Node Public IP must use public IP prefixes, and these prefixes need to be added as authorized ranges

1

2

.

To resolve this, you can follow these steps:

1. Create a Public IP Prefix:
2. az network public-ip prefix create --length 28 --location <region> --name <publicIPPrefixName> --resource-group <resourceGroup>
3. Add the Node Pool with the Public IP Prefix:

az aks nodepool add --resource-group <resourceGroup> --cluster-name <aksClusterName> --name <newNodePool> --enable-node-public-ip --node-public-ip-prefix-id /subscriptions/<subscription-id>/resourceGroups/<resourceGroup>/providers/Microsoft.Network/publicIPPrefixes/<publicIPPrefixName>

Issue 2- Not able to add ip prefix, because I need to add min 28 ip for that but my subscription does not allow that.

PS /home/parthasarathi> az network public-ip prefix create --length 28 --location eastus --name publicIPPrefix --resource-group Nacha-Test

(IPv4StandardSkuPublicIpCountLimitReached) Cannot create more than 10 IPv4 Standard SKU public IP addresses for this subscription in this region.

Code: IPv4StandardSkuPublicIpCountLimitReached

Message: Cannot create more than 10 IPv4 Standard SKU public IP addresses for this subscription in this region.

Resolution –

Make public cluster and add node pool

A screenshot of a computer

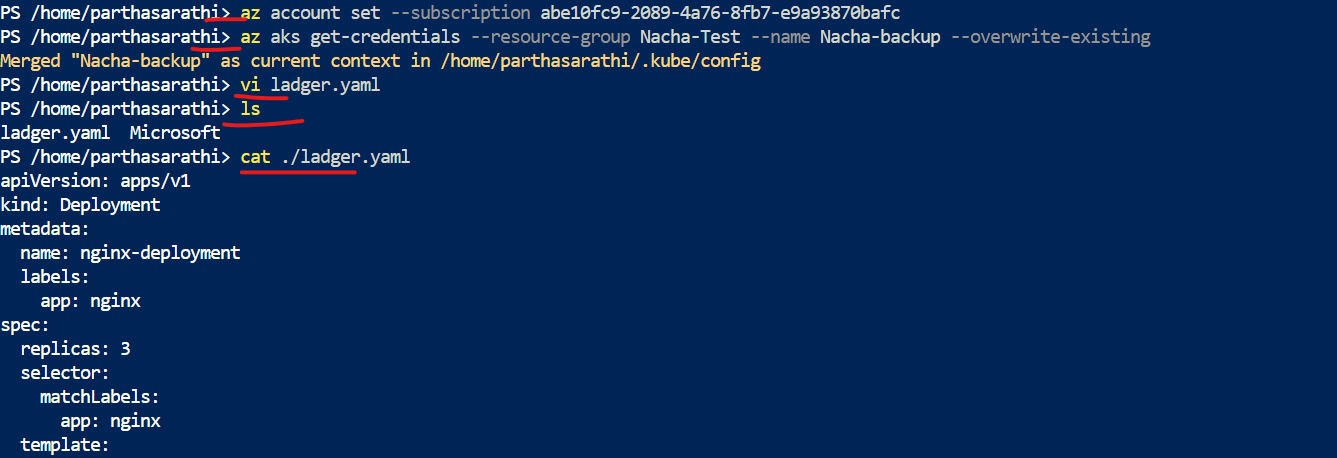
Description automatically generated

Added Node pool

A screenshot of a computer

Description automatically generated

Connect to cluster and deployment to deploy pod



Created deployment file o cloud and deployed.

Yaml file—

apiVersion: v1

kind: Namespace

metadata:

name: test

---

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-deployment

namespace: test

labels:

app: nginx

spec:

replicas: 3

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

nodeSelector:

agentpool: phixiusprod

containers:

- name: nginx

image: nginx:latest

ports:

- containerPort: 80

---

apiVersion: v1

kind: Service

metadata:

name: nginx-service

namespace: test

spec:

selector:

app: nginx

ports:

- protocol: TCP

port: 80

targetPort: 80

type: LoadBalancer

We deployed under test namespace

A screen shot of a computer program

Description automatically generated

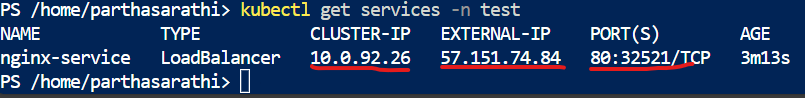
How to access application

ince the service is of type LoadBalancer, it will provision an external IP address that you can use to access your NGINX application.

To find the external IP address, you can run the following command:

kubectl get services -n test

This will list all the services in the test namespace, including the nginx-service. Look for the EXTERNAL-IP column in the output. Once the external IP address is assigned (it might take a few minutes), you can access your NGINX application by navigating to that IP address in your web browser.



A screenshot of a computer

Description automatically generated

Task-

Given cluster, storage and vault access to Sukhdeep.

Check pods are running on specified node pool “phixius prod”

A blue screen with white text

Description automatically generated

Task-

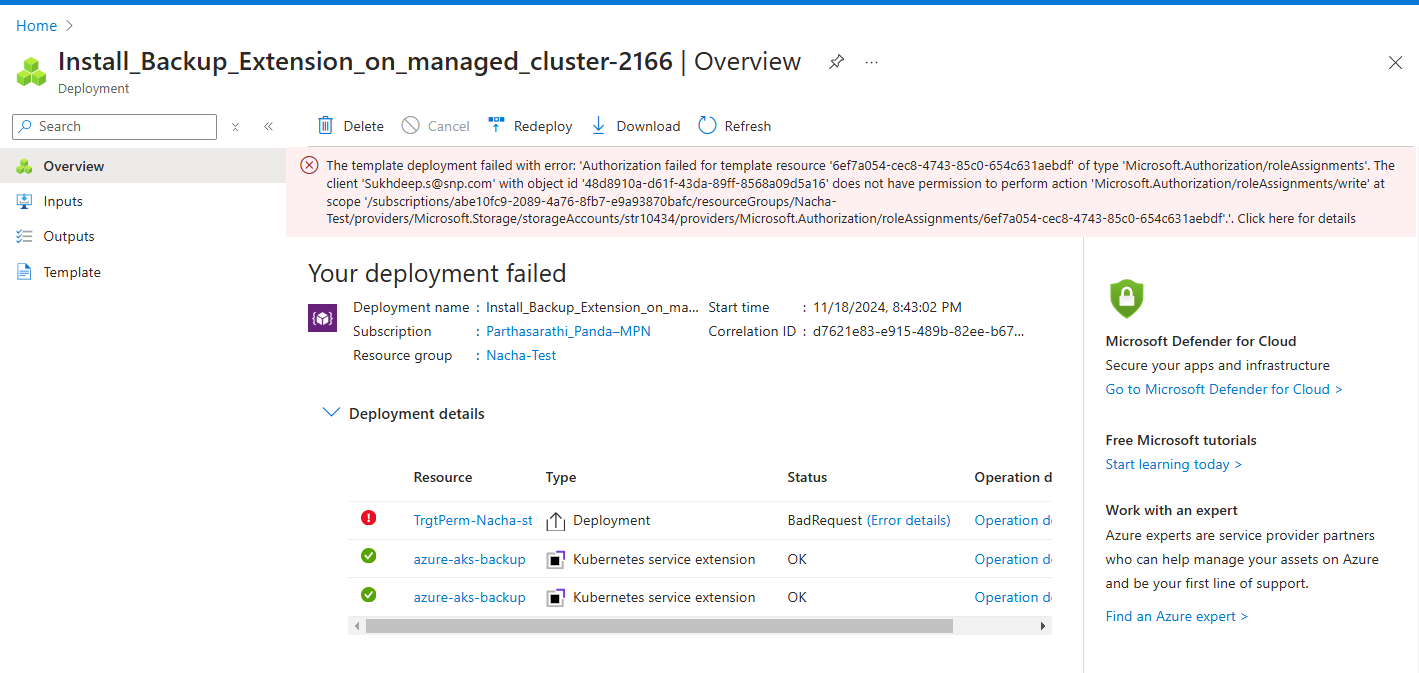
Sukhdeep will install agent for backup on cluster

A screenshot of a computer

Description automatically generated

Note- alse need to provide reader access on str rg and mc rg.

Issue-



error –

The template deployment failed with error: 'Authorization failed for template resource '6ef7a054-cec8-4743-85c0-654c631aebdf' of type 'Microsoft.Authorization/roleAssignments'. The client 'Sukhdeep.s@snp.com' with object id '48d8910a-d61f-43da-89ff-8568a09d5a16' does not have permission to perform action 'Microsoft.Authorization/roleAssignments/write' at scope '/subscriptions/abe10fc9-2089-4a76-8fb7-e9a93870bafc/resourceGroups/Nacha-Test/providers/Microsoft.Storage/storageAccounts/str10434/providers/Microsoft.Authorization/roleAssignments/6ef7a054-cec8-4743-85c0-654c631aebdf'.'. (Code: InvalidTemplateDeployment)

Reason-

Because the account Sukhdeep have contributor role not the admin, to perform this activity we need owner role because the person should be able to assign role for resources.

A screenshot of a computer

Description automatically generated

But the agent will be installed.

A white screen with text

Description automatically generated with medium confidence

Mistake –

We ignored that and proceed to next as the agent was installed.

Next-

We went to configured backup.

A screenshot of a computer

Description automatically generated

Issue- access issue for the user he need owner access.

Next – I did the task as owner

A screenshot of a computer

Description automatically generated

Prechecks for aks Microsoft doc-

[Azure Kubernetes Service (AKS) backup using Azure Backup prerequisites - Azure Backup | Microsoft Learn](https://learn.microsoft.com/en-us/azure/backup/azure-kubernetes-service-cluster-backup-concept)

Issue-

Please fix errors before continuing.

A screenshot of a computer

Description automatically generated

But everything looks good.

Resolution reinstall ihe agent on primary cluster and refresh the cluster

Cmd to refresh the cluster

>> az resource update --name xxxxxx --namespace Microsoft.ContainerService  --resource-group xxxx --resource-type ManagedClusters  --subscription xxxxx

A blue screen with white text

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Once update done reinstall the agent-

A screenshot of a computer

Description automatically generated

Retry Backup-

It was successfully done.

A screenshot of a computer

Description automatically generated

NOTE- the agent installation can have many types of issue, issue to configure backup (our sceniaro). Isste to restore (nacha production cluster issue)

Run Backup-

A screenshot of a computer

Description automatically generated

Note- backup most probably takes some time. For nacha it was not configured properly so the backup got completed in 2 min with corrupted data.

A screenshot of a computer

Description automatically generated

Restore

A screenshot of a computer

Description automatically generated

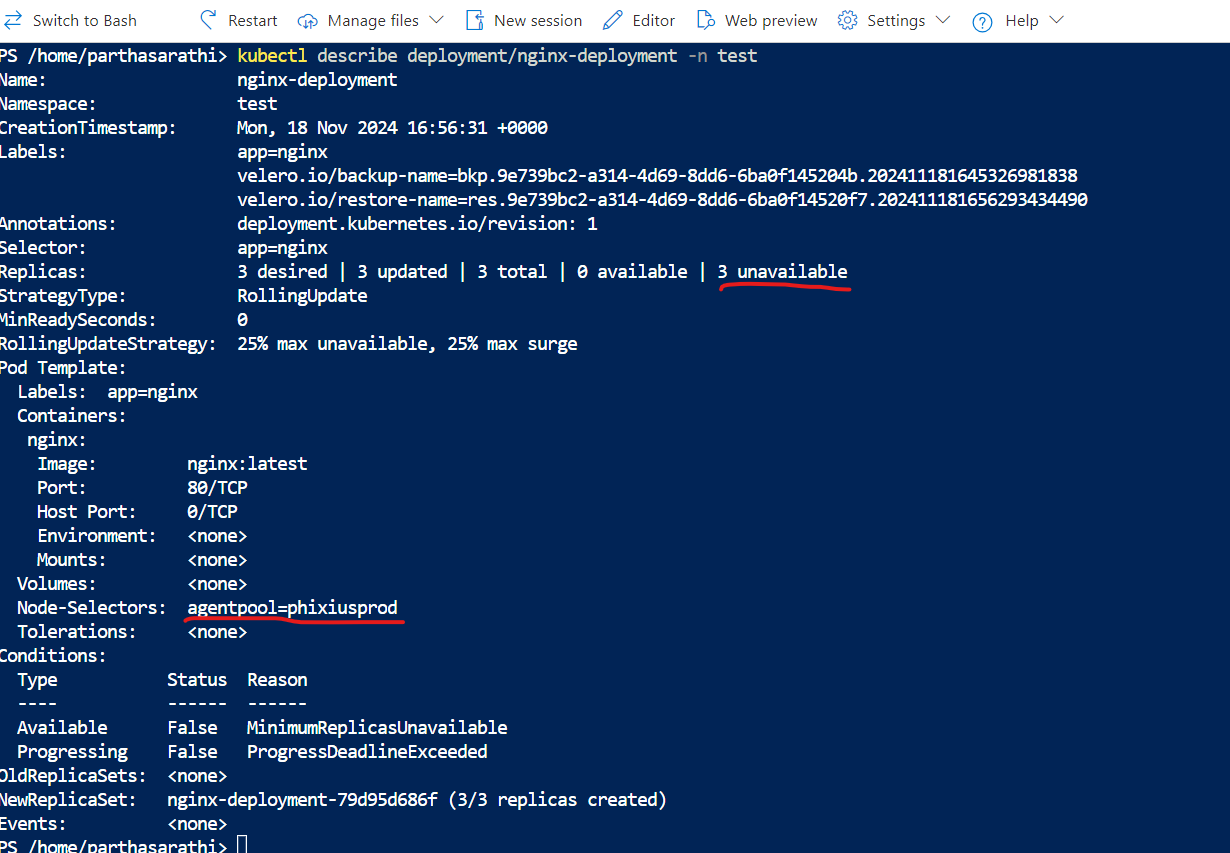
Triggered restoration on a new cluster but that cluster only have one nodepool which is systempool but the deployment on actual cluster configured to deploy on a particular nodepool “phixius prod”

Result pods will not get schedule because it’s unable to find agentpool=phixiusprod.

A blue screen with white text

Description automatically generated

We will also be not able to see any logs on pos



Resolution-

Create required nodepool on restoration cluster.

A screenshot of a computer

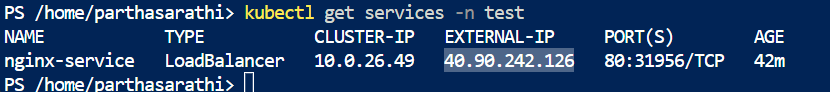
Description automatically generated

Post creation of the node pool pods automatically got scheduled on targeted node and started running fine.

A screen shot of a computer program

Description automatically generated

Let’s check application accessibility.

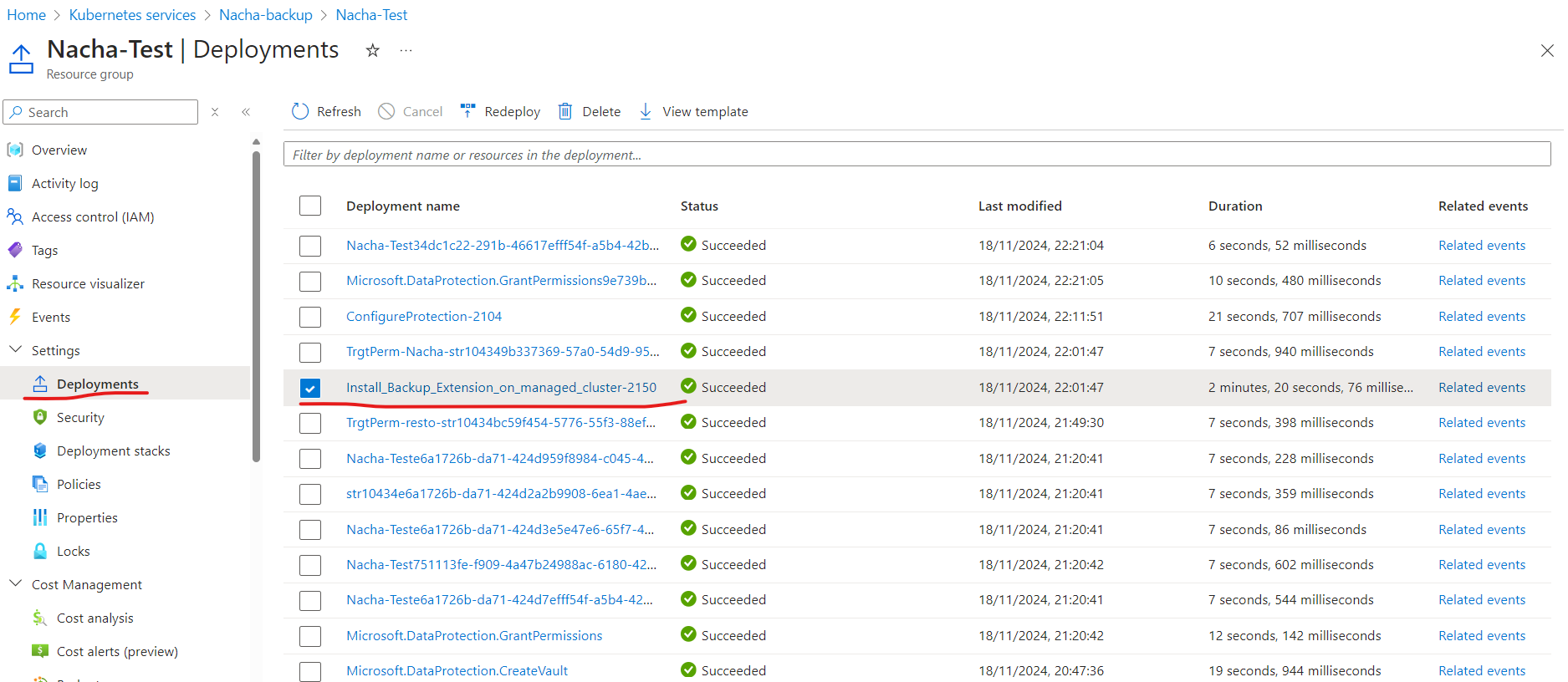


A screenshot of a computer

Description automatically generated

Action For Nacha-

Cleanup the installed agent o both of the cluster, refresh the cluster recreate backup instance and reconfigure backup.  
  
Target we are going to suggest then to do it by script.  
  
How to do that.

Lets delete the backup triggered the agent installation, uninstall old agent and re install.  
  
Go to rg where the deployment was done. Download the deployment file.  
  


Go for it and download  
  
A screenshot of a computer

Description automatically generated

Deleting extension to reinstall and confirm

A close up of a text

Description automatically generated

Go to cloudshell and upload files

Such as deployment and deployment operations. Which we downloaded from deployment

A screenshot of a computer

Description automatically generated

A blue background with white text

Description automatically generated

Run bash-

az deployment group create \

--resource-group <ResourceGroupName> \

--template-file ./deployment.json \

--parameters ./deployment\_operations.json

A screenshot of a computer

Description automatically generated  
  
Note – in my case I did rename for files. That’s why bash contains different name.

But if I did the operation and downloaded file, and someone else is executing it’s not gonna work. Need to modify.