



Kauno technologijos universitetas

Informatikos fakultetas

P175B120 Paslaugų programavimas debesų kompiuterijoje

Laboratorinis darbas Nr. 3

Nedas Liaudanskis IFF-1/9

Studentas

dėst. Pilkauskas Vytautas

Dėstytojas

Turinys

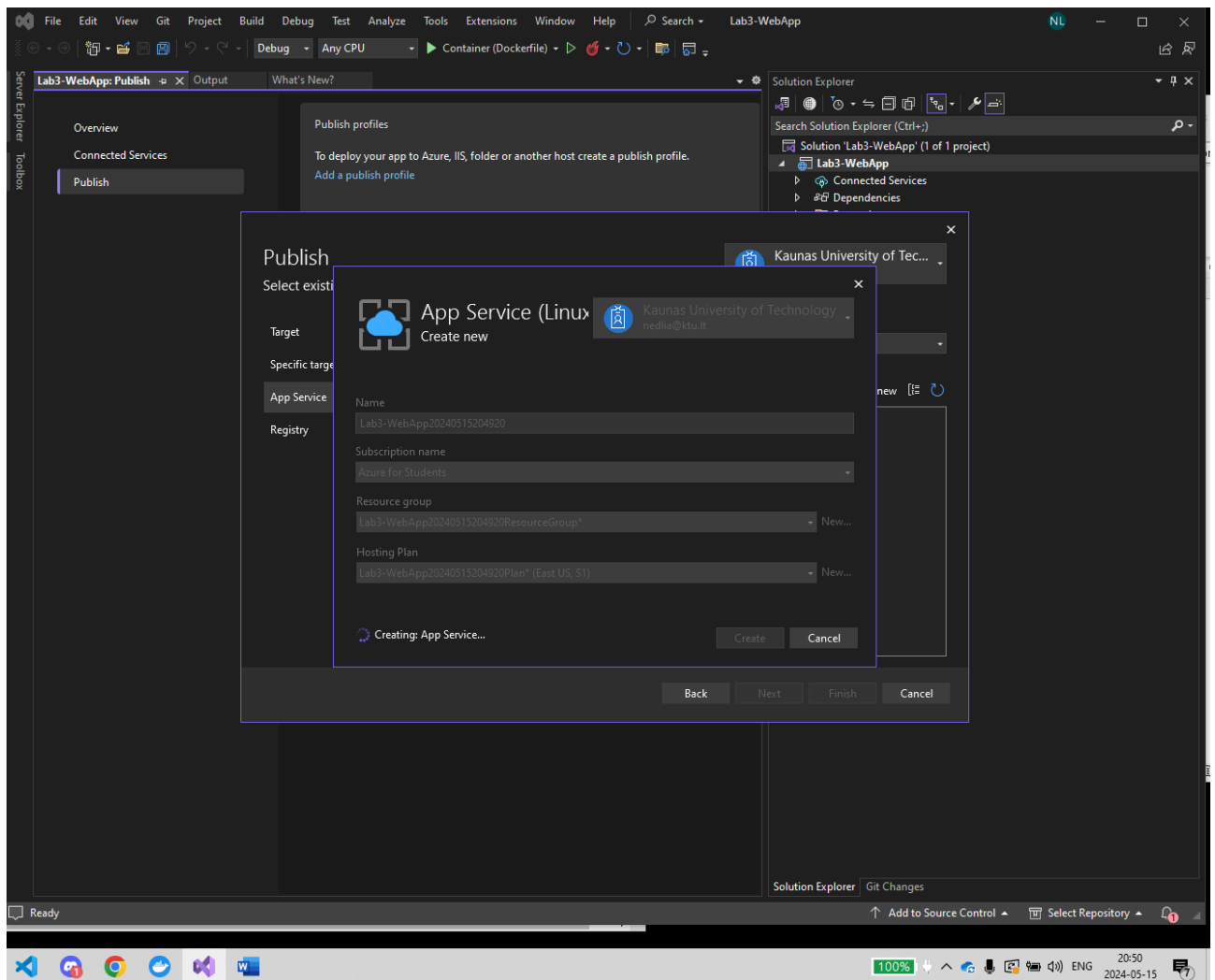
Lab 3.1 Dalis	3
3.2 Dalis	7

Paveikslėlių sąrašas

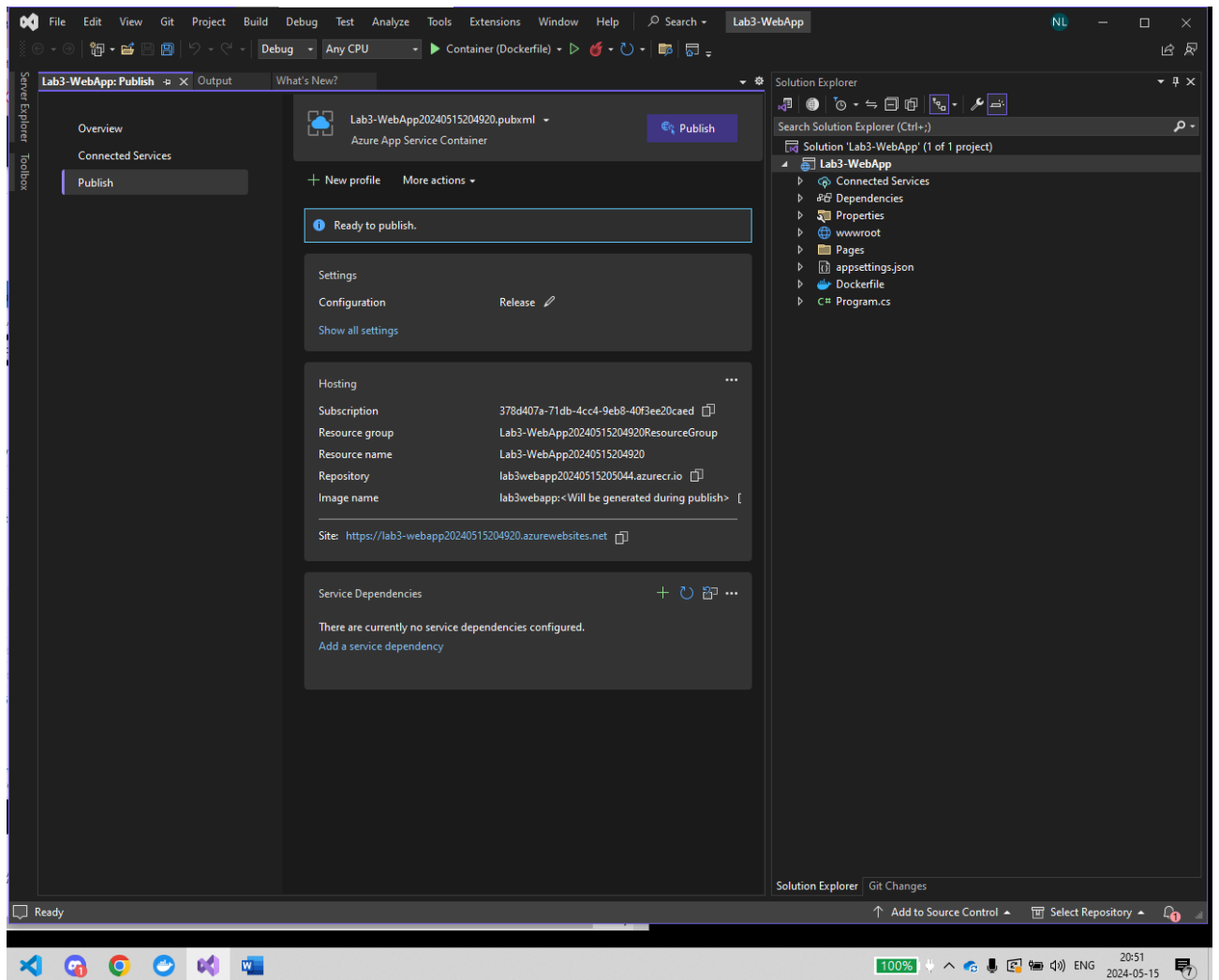
pav. 1 Programos konteinerio sukūrimas	3
pav. 2 Konteinerio registro sukūrimas	4
pav. 3 Programa perkeliama į Azure App Service	5
pav. 4 Sukurtas konteinerio registras ir Container instance	6
pav. 5 Programa esanti Azure Container Instance	6
pav. 6 Paleista programa	7
pav. 7 Sugeneruoti resursai	8
pav. 8 Pavaizduoti AKS klasterio mazgai	8
pav. 9 Pavaizduota stotelė	8
pav. 10 Pavaizduotas AKS klasterio informaciją	9
pav. 11 Darbo eiga, modifikuojant ingress.yaml failą	9
pav. 12 Viso AKS klasterio komponentų atvaizdavimas	9
pav. 13 Svetainės vaizdas	10
pav. 14 AKS klasteris	10

Lab 3.1 Dalis

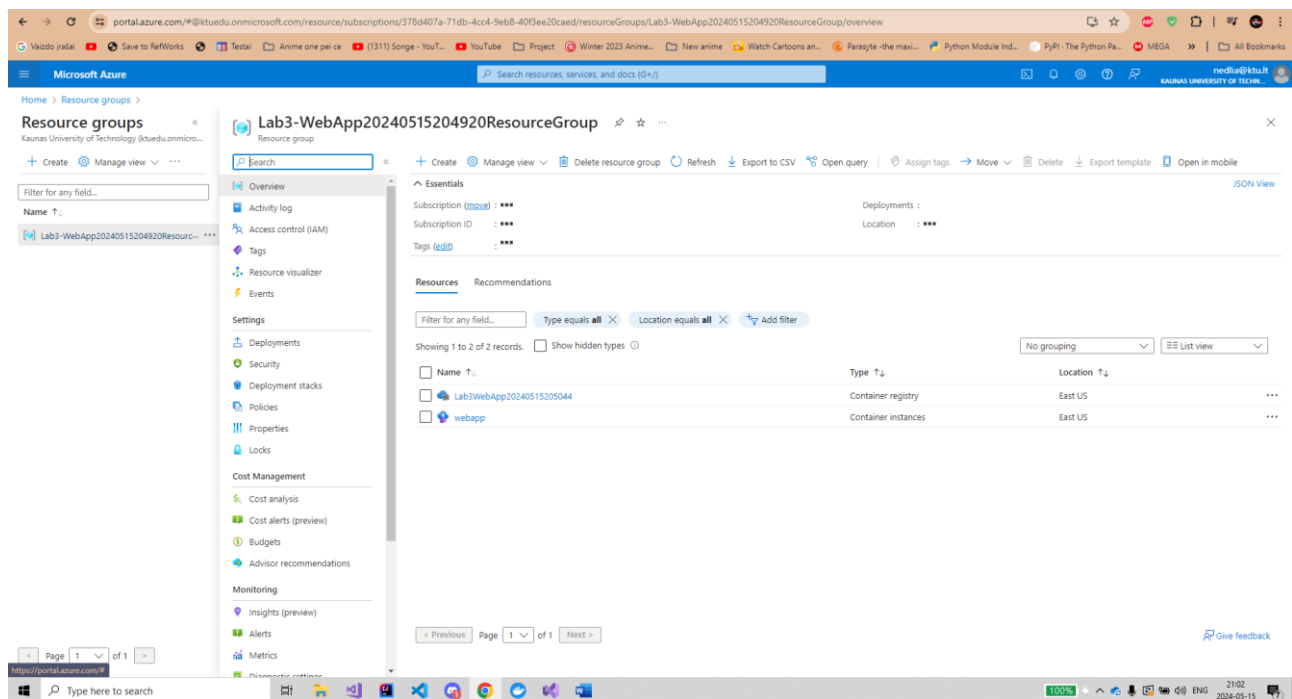
Sukurta žiniatinklio ASP.NET programa, visual studio programoje. Programa įdedama kaip docker konteineris į „Azure Container Registry“ ir paleidžiama per „Azure App Service“. Tada programa perkiamama ir paleidžiama kaip „Azure Container Instance“.



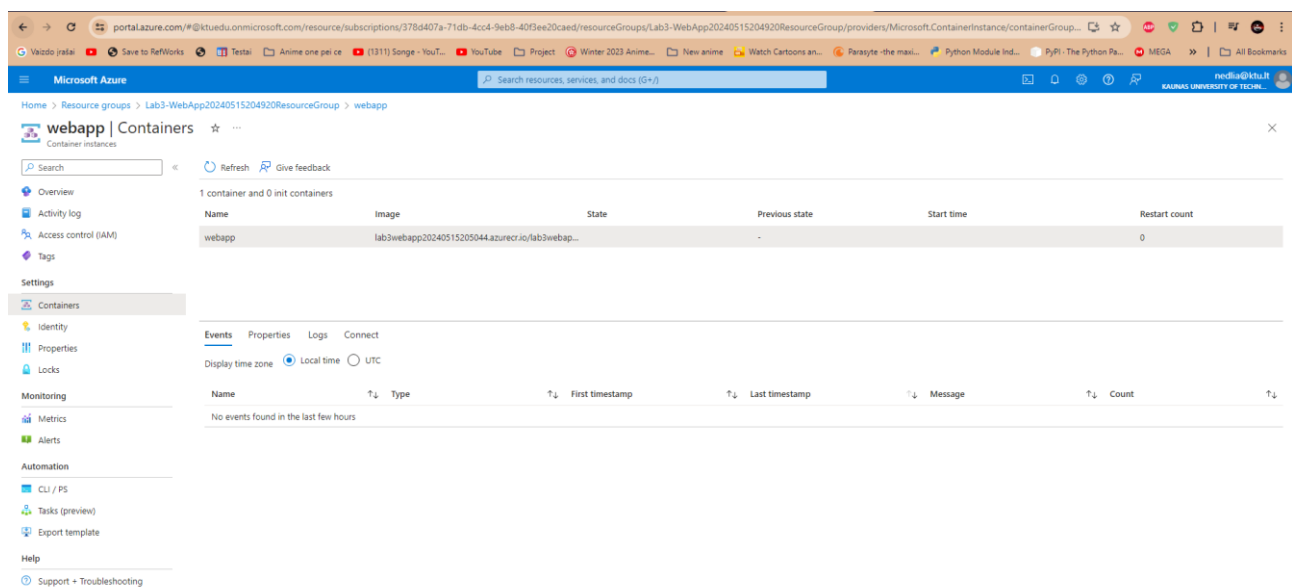
pav. 1 Programos konteinerio sukūrimas



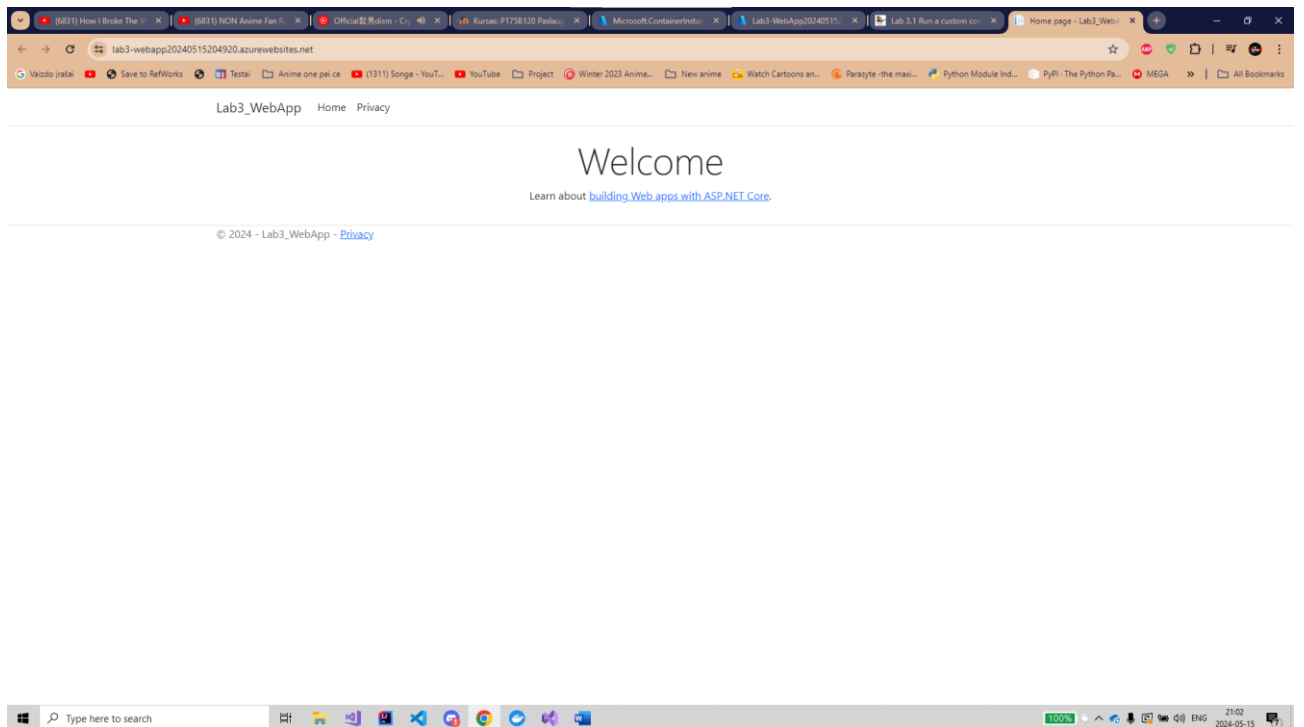
pav. 3 Programa perkeliama į Azure App Service



pav. 4 Sukurtas konteinerio registras ir Container instance



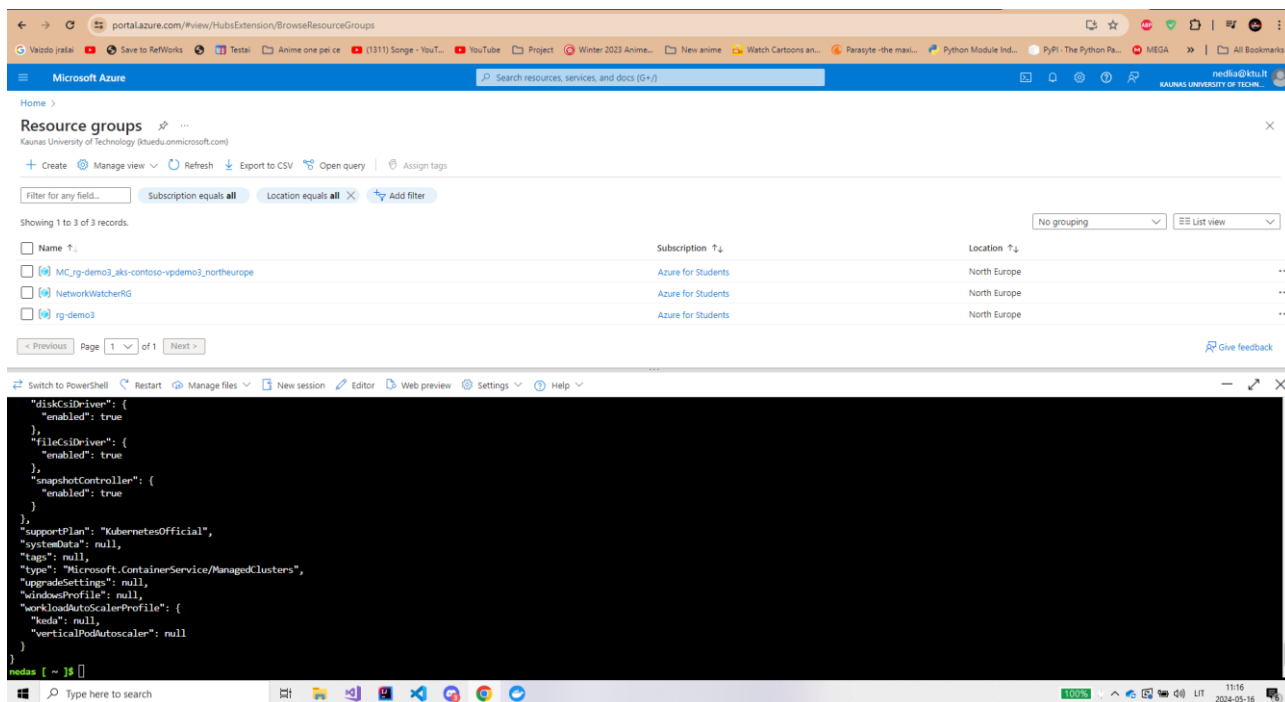
pav. 5 Programa esanti Azure Container Instance



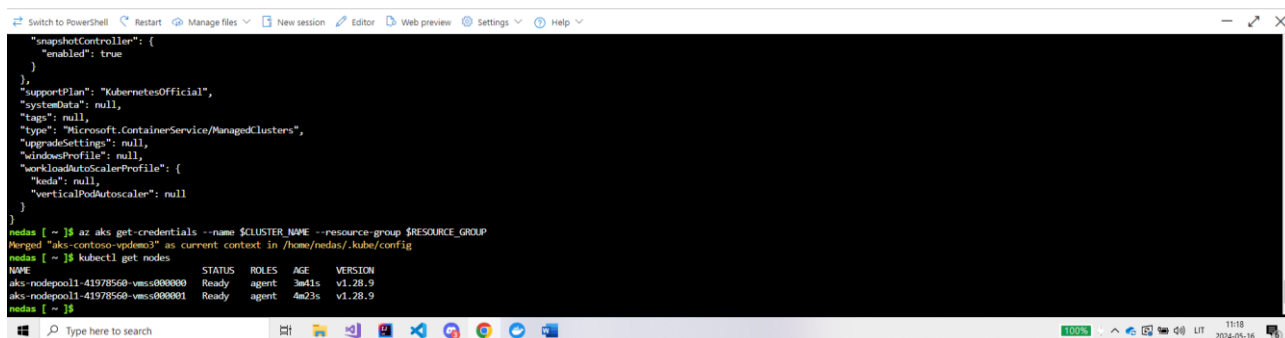
pav. 6 Paleista programa

3.2 Dalis

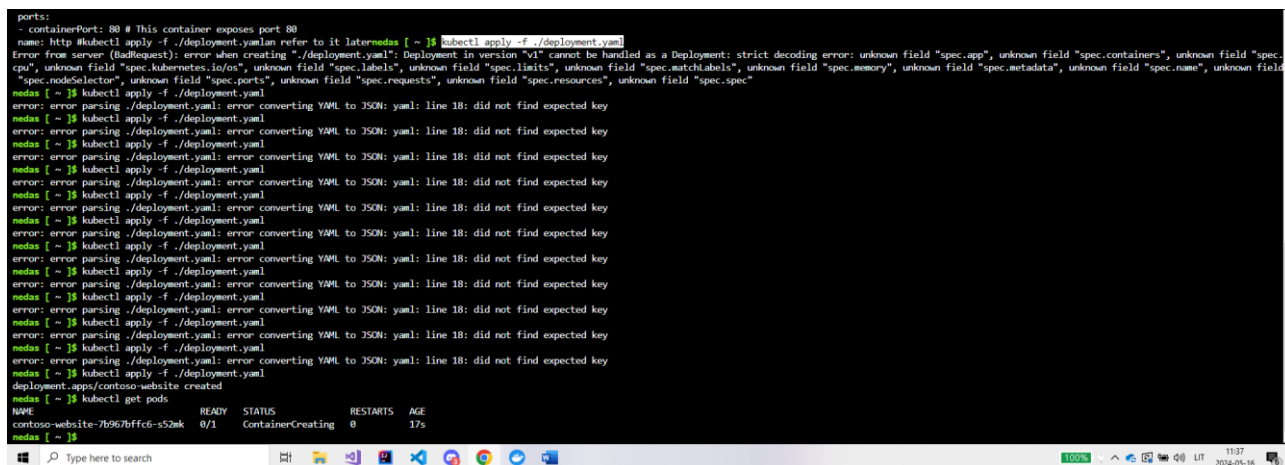
Sukuriamas, AKS klasteris Azure portale, naudojant cloud shell bash. Pirmiausiai sukuriamas nauja resursų grupė, tada pridėdame AKS ir jame esančius mazgus. Patikriname ar jie veikia teisingai su komanda „kubectl get nodes“. Tada aprašome manifestacijos failą, kuriame pateikiama pagrindinė klasterio informacija. Tada galime patikrini stoteles naudojant šią komandą: „kubectl get pods“. Dar lieka sukonfiguruoti „ingress.yaml“ ir „service.yaml“ failus, kurie aprašo mūsų svetainę. Ir po viso šito, mes savo svetainę perkėlėme į AKS klasteri.



pav. 7 Sugeneruoti resursai



pav. 8 Pavaizduoti AKS klasterio mazgai



pav. 9 Pavaizduota stotelė


```
deployment.apps/cotoso-website created
nadas [ ~ ]$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
cotoso-website-7b967bffc6-s52mk    0/1     ContainerCreating   0          17s
nadas [ ~ ]$ touch service.yaml
nadas [ ~ ]$ kubectl apply -f ./service.yaml
service/cotoso-website created
nadas [ ~ ]$ kubectl get service
NAME      TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
cotoso-website  ClusterIP   10.0.171.82   <none>        80/TCP    25s
kubernetes    ClusterIP   10.0.0.1     <none>        443/TCP   29m
nadas [ ~ ]$
```

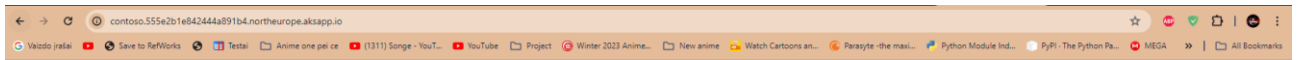
pav. 10 Pavaizduotas AKS klasterio informaciją

```
Bash
FILES
+ azure
+ azure
+ cache
+ kube
+ local
+ ash
+ Microsoft
+ bash_history
+ bash_logout
+ bash_profile
+ bashrc
+ tmux.conf
+ zshrc
+ deployment.yaml
+ ingress.yaml
+ ingress.yaml
1 apiVersion: networking.k8s.io/v1
2 kind: Ingress
3 metadata:
4   name: cotoso-website
5   annotations:
6     kubernetes.io/ingress.class: addon-http-application-routing
7 spec:
8   rules:
9     - host: cotoso.vdemo3.northeurope.aksapp.io
10      http:
11        paths:
12          - path: / # Which path is this rule referring to
13            pathType: Prefix
14            backend: # How the Ingress will handle the requests
15              service:
16                name: cotoso-website # Which service the request will be forwarded
17                port:
18                  name: http # Which port in that service
nadas [ ~ ]$ az network dns zone list
an alphanumeric character (e.g. 'example.com', regex used for validation is '[a-z0-9]([-a-z0-9]*[a-z0-9])?(\.?[a-z0-9]([-a-z0-9]*[a-z0-9])?)*')
[
  {
    "etag": "d3df06ca-4bf7-47d8-8270-b5c1610e160",
    "id": "/subscriptions/378d407a-71db-4cc4-9eb8-40f3ee20caed/resourceGroups/mc_rg-demo3_aks-cotoso-vdemo3_northeurope/providers/Microsoft.Network/dnszones/555e2b1e84244a891b4.northeurope.aksapp.io",
    "location": "global",
    "maxNumberOfRecordSets": 10000,
    "name": "555e2b1e84244a891b4.northeurope.aksapp.io",
    "nameServers": [
      "ns1-37.azure-dns.com.",
      "ns2-37.azure-dns.net.",
      "ns3-37.azure-dns.org.",
      "ns4-37.azure-dns.info."
    ],
    "numberOfRecordSets": 2,
    "resourceGroup": "mc_rg-demo3_aks-cotoso-vdemo3_northeurope",
    "tags": {},
    "type": "Microsoft.Network/dnszones",
    "zoneType": "Public"
  }
]
nadas [ ~ ]$ kubectl apply -f ./ingress.yaml
The Ingress "cotoso-website" is invalid: spec.rules[0].host: Invalid value: "cotoso.555e2b1e84244a891b4.<northeurope>.aksapp.io": a lowercase RFC 1123 subdomain must consist of lower case alphanumeric characters, '-' or '.', and must
```

pav. 11 Darbo eiga, modifikuojant ingress.yaml failą

```
"ns2-37.azure-dns.net.",
"ns3-37.azure-dns.org.",
"ns4-37.azure-dns.info."
],
"numberOfRecordSets": 2,
"resourceGroup": "mc_rg-demo3_aks-cotoso-vdemo3_northeurope",
"tags": {},
"type": "Microsoft.Network/dnszones",
"zoneType": "Public"
}
]
nadas [ ~ ]$ kubectl apply -f ./ingress.yaml
ingress.networking.k8s.io/cotoso-website configured
nadas [ ~ ]$ kubectl get ingress cotoso-website
NAME      CLASS      HOSTS                                ADDRESS      PORTS      AGE
cotoso-website  <none>    cotoso.555e2b1e84244a891b4.northeurope.aksapp.io  172.205.35.15  80        4m22s
nadas [ ~ ]$ kubectl get service cotoso-website
NAME      TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
cotoso-website  ClusterIP   10.0.171.82   <none>        80/TCP    13m
nadas [ ~ ]$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
cotoso-website-7b967bffc6-s52mk    1/1     Running   0          19m
nadas [ ~ ]$ kubectl get nodes
NAME                                STATUS   ROLES    AGE   VERSION
aks-nodepool1-41978560-vmss000000  Ready   agent    42m   v1.28.9
aks-nodepool1-41978560-vmss000001  Ready   agent    42m   v1.28.9
nadas [ ~ ]$
```

pav. 12 Viso AKS klasterio komponentų atvaizdavimas



This site can't be reached

Check if there is a typo in contoso.555e2b1e84244a891b4.northeurope.aksapp.io.

If spelling is correct, try [running Windows Network Diagnostics](#).

DNS_PROBE_FINISHED_NXDOMAIN

Reload

pav. 13 Svetainės vaizdas

The screenshot shows the Azure portal interface for a Kubernetes service named 'aks-contoso-vpdemo3'. The left sidebar contains navigation options like Overview, Activity log, Access control, Tags, and various Kubernetes resources. The main content area is divided into several sections:

- Essentials:** Provides key information about the cluster, including its resource group ('rg-demo3'), status ('Succeeded (Running)'), subscription ('Azure for Students'), location ('North Europe'), and subscription ID.
- Node pools:** Details the configuration of the node pools, showing one pool with Kubernetes version 1.28.9 and Standard_B2s node sizes.
- Configuration:** Lists various cluster settings such as Kubernetes version (1.28.9), auto upgrade type, automatic upgrade scheduler, node security channel type, and authentication/authorization settings.
- Networking:** Displays network-related parameters like API server address, network type (kubernetes), pod CIDR, service CIDR, and DNS service IP.
- Integrations:** Shows the status of integrations with other services, such as container insights and workspace resource ID.

pav. 14 AKS klasteris