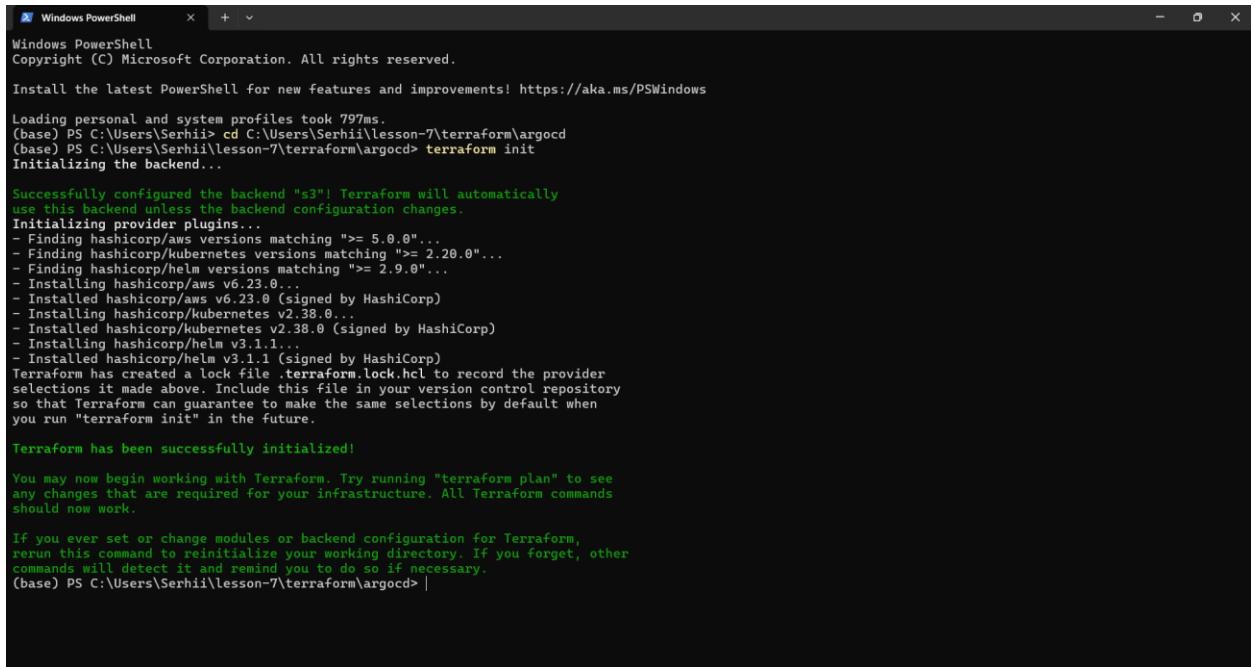


## Terraform init



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

Loading personal and system profiles took 797ms.
(base) PS C:\Users\Serhii> cd C:\Users\Serhii\lesson-7\terraform\argocd
(base) PS C:\Users\Serhii\lesson-7\terraform\argocd> terraform init
Initializing the backend...
Successfully configured the backend "s3"! Terraform will automatically
use this backend unless the backend configuration changes.
Initializing provider plugins...
- Finding hashicorp/aws versions matching ">= 5.0.0"...
- Finding hashicorp/kubernetes versions matching ">= 2.20.0"...
- Finding hashicorp/helm versions matching ">= 2.9.0"...
- Installing hashicorp/aws v6.23.0...
- Installed hashicorp/aws v6.23.0 (signed by HashiCorp)
- Installing hashicorp/kubernetes v2.38.0...
- Installed hashicorp/kubernetes v2.38.0 (signed by HashiCorp)
- Installing hashicorp/helm v3.1.1...
- Installed hashicorp/helm v3.1.1 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
(base) PS C:\Users\Serhii\lesson-7\terraform\argocd>
```

Для реалізації задачі було змінено налаштування eks-кластеру з попереднього д/з

```
eks_managed_node_groups = {

    cpu_nodes = {

        instance_types = ["t3.small"]

        desired_size  = 3

        max_size      = 3

        min_size      = 1

    }

}
```

Запущено ArgoCD

```
terraform apply -auto-approve
```

```
| Windows PowerShell | + - x
# kubernetes_namespace.infra_tools will be created
+ resource "kubernetes_namespace" "infra_tools" {
  + id                      = (known after apply)
  + wait_for_default_service_account = false
  + metadata {
    + generation      = (known after apply)
    + name            = "infra-tools"
    + resource_version = (known after apply)
    + uid             = (known after apply)
  }
}

Plan: 2 to add, 0 to change, 0 to destroy.
kubernetes_namespace.infra_tools: Creating...
kubernetes_namespace.infra_tools: Creation complete after 1s [id=infra-tools]
helm_release.argocd: Creating...
helm_release.argocd: Still creating... [00m10s elapsed]
helm_release.argocd: Still creating... [00m20s elapsed]
helm_release.argocd: Still creating... [00m30s elapsed]
helm_release.argocd: Still creating... [00m40s elapsed]
helm_release.argocd: Still creating... [00m50s elapsed]
helm_release.argocd: Still creating... [01m00s elapsed]
helm_release.argocd: Creation complete after 1m2s [id=argocd]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:

argo_namespace = "infra-tools"
argo_server = "argocd-server.infra-tools.svc.cluster.local"
argocd_namespace = "infra-tools"
(base) PS C:\Users\Serhii\lesson-7\terraform\argocd> kubectl get pods -n infra-tools
NAME                           READY   STATUS    RESTARTS   AGE
argocd-application-controller-0  1/1     Running   0          95s
argocd-applicationset-controller-5bc84845bc-jrcvc  1/1     Running   0          96s
argocd-notifications-controller-7f5b9964c5-6dmq7  1/1     Running   0          96s
argocd-redis-f98ffef9cd-29cjf  1/1     Running   0          95s
argocd-repo-server-6694b8b5d-jx6th  1/1     Running   0          95s
argocd-server-7d46898d85-k5rmh  1/1     Running   0          95s
(base) PS C:\Users\Serhii\lesson-7\terraform\argocd> |
```

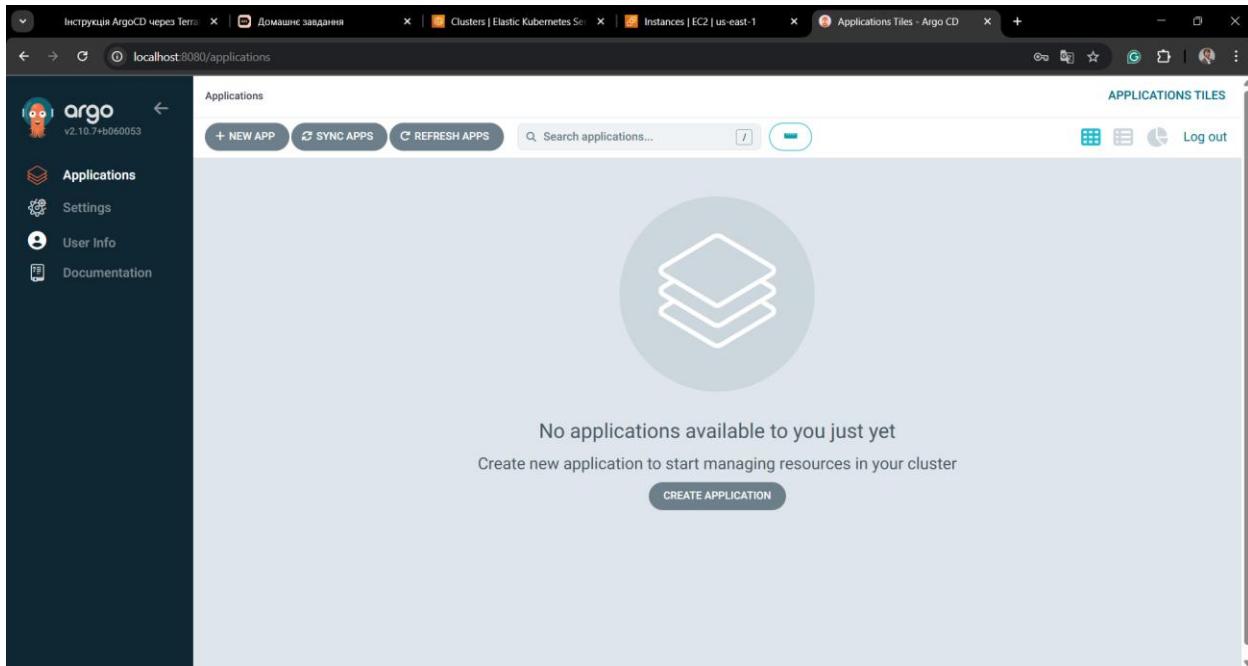
Отримуємо пароль (логін по замовчуванню admin)

```
kubectl -n infra-tools get secret argocd-initial-admin-secret -o jsonpath="{.data.password}" > enc.txt
```

```
certutil -decode enc.txt dec.txt
```

```
type dec.txt
```

Отримали доступ до web-інтерфейсу ArgoCD



## Створення застосунку «nginx-app», який вибрано для виконання д/з

```
| Командний рядок
Error from server (NotFound): namespaces "application" not found
C:\Users\Serhii\lesson-7\goit-argo>kubectl apply -f application.yaml -n infra-tools
application.argoproj.io/nginx-app created

C:\Users\Serhii\lesson-7\goit-argo>kubectl delete application nginx-app -n infra-tools
application.argoproj.io "nginx-app" deleted

C:\Users\Serhii\lesson-7\goit-argo>kubectl apply -f application.yaml -n infra-tools
application.argoproj.io/nginx-app created

C:\Users\Serhii\lesson-7\goit-argo>kubectl get pods -n application
NAME          READY   STATUS    RESTARTS   AGE
nginx-app-6ccb498564-6x55b   0/1     ImagePullBackOff   0      4m42s

C:\Users\Serhii\lesson-7\goit-argo>kubectl delete application nginx-app -n infra-tools
application.argoproj.io "nginx-app" deleted

C:\Users\Serhii\lesson-7\goit-argo>kubectl delete namespace application
namespace "application" deleted

C:\Users\Serhii\lesson-7\goit-argo>kubectl apply -f application.yaml -n infra-tools
application.argoproj.io/nginx-app created

C:\Users\Serhii\lesson-7\goit-argo>kubectl get pods -n application
NAME          READY   STATUS    RESTARTS   AGE
nginx-app-ingress-nginx-admission-create-bz8d9   0/1     Completed   0      18s
nginx-app-ingress-nginx-controller-6b887844c7-wbpqp  0/1     Running   0      9s

C:\Users\Serhii\lesson-7\goit-argo>kubectl get pods -n application
NAME          READY   STATUS    RESTARTS   AGE
nginx-app-ingress-nginx-controller-6b887844c7-wbpqp  1/1     Running   0      3m35s

C:\Users\Serhii\lesson-7\goit-argo|
```

Результати роботи nginx-app в argoCD

The screenshot shows the Argo CD web interface for managing a Kubernetes application named "nginx-app". The left sidebar contains navigation links for "Applications", "Settings", "User Info", and "Documentation". The main area displays the "nginx-app" application details. At the top, there are tabs for "DETAILS", "DIFF", "SYNC", "SYNC STATUS", "HISTORY AND ROLLBACK", "DELETE", and "REFRESH". The "SYNC STATUS" section indicates the application is "Synced" to version 4.10.0 (4.10.0). Below this, a message states "Auto sync is enabled." On the right, a "LAST SYNC" section shows a "Sync OK" status with a timestamp of "Succeeded a minute ago (Mon Dec 01 2025 22:12:06 GMT+0200)". A large central panel titled "APPLICATION DETAILS TREE" shows a hierarchical sync tree for the "nginx-app" application. The tree starts with a single "nginx-app" node at the bottom, which branches into multiple "nginx-app-ingress" nodes. These further branch into "nginx-ingress-controller" and "nginx-ingress-routes" nodes. The entire sync tree is shown with green circular icons, indicating a healthy sync status across all components.

The screenshot shows the Argo CD web interface at [localhost:8080/applications/infra-tools/nginx-app?resource=&view=network](http://localhost:8080/applications/infra-tools/nginx-app?resource=&view=network). The left sidebar has sections for Applications, Settings, User Info, and Documentation. The main area shows the 'nginx-app' application details. The 'SYNC STATUS' section indicates it is 'Synced' to version 4.10.0 (4.10.0) with a green checkmark and the message 'Auto sync is enabled.' Below this is a network diagram showing the flow from a cloud icon to a deployment (a2e4283ec00b04f4eb72f8b8), which then points to two service entries ('nginx-app-ingress-nginx-contr...' and another 'nginx-app-ingress-nginx-contr...'), and finally to a pod ('nginx-app-ingress-nginx-contr...'). Each component is shown with a green heart icon and the status 'running'. A tooltip for the second service entry says '(a few seconds)'.