**What is a NetworkPolicy in OpenShift?**   
  
A NetworkPolicy is a Kubernetes/OpenShift resource that controls traffic flow between pods, namespaces, and external resources.   
  
**It defines rules (allow only) for:**   
Ingress: Traffic coming into the pod   
Egress: Traffic going out of the pod

**Default behavior in OpenShift:**   
When you create a NetworkPolicy, only the traffic explicitly allowed is permitted, and everything else is denied.

**Why Use NetworkPolicy?**   
Security   
Prevent unauthorized pod-to-pod communication   
Isolate workloads (e.g., dev vs prod)   
  
**How Does It Work in OpenShift?**   
OpenShift uses a CNI plugin (e.g., OpenShift SDN or OVN-Kubernetes) to enforce network policies.

**Here's what happens behind the scenes**:   
Pods in a Namespace are "open" by default (all can talk to each other)   
As soon as a NetworkPolicy is applied, communication is blocked by default   
You explicitly whitelist traffic you want to allow   
The underlying CNI plugin (e.g., OVN-K) enforces these rules in iptables/OVS   
  
**What is an Ingress Rule?**   
Controls what traffic is allowed INTO a pod.   
  
**What is an Egress Rule?**   
Controls where a pod is allowed to send outbound traffic

**Scenario Setup**   
You have two pods:   
frontend pod (label: app=frontend)   
backend pod (label: app=backend)   
  
**Part 1: Ingress Rule Example**   
Allow only frontend to access backend   
Other pods (even in the same namespace) should not be able to talk to the backend.   
  
apiVersion: [networking.k8s.io/v1](https://web.telegram.org/a/networking.k8s.io/v1)   
kind: NetworkPolicy   
metadata:   
name: allow-frontend-to-backend   
namespace: myapp   
spec:   
podSelector:   
matchLabels:   
app: backend # Target: backend pod   
ingress:   
- from:   
- podSelector:   
matchLabels:   
app: frontend # Only frontend pod allowed to access   
  
Explanation:   
This policy applies to the backend pod   
It only allows ingress (incoming) traffic from pods with label app=frontend   
Any other pod trying to reach backend is blocked   
  
**Part 2: Egress Rule Example**   
Allow backend pod to make external HTTP requests only (port 80)   
apiVersion: [networking.k8s.io/v1](https://web.telegram.org/a/networking.k8s.io/v1)   
kind: NetworkPolicy   
metadata:   
name: allow-backend-egress-http   
namespace: myapp   
spec:   
podSelector:   
matchLabels:   
app: backend # Apply policy to backend pod   
policyTypes:   
- Egress   
egress:   
- to:   
- ipBlock:   
cidr: [0.0.0.0/0](https://web.telegram.org/a/0.0.0.0/0)      # Allow to any IP (internet)   
ports:   
- protocol: TCP   
port: 80 # Only HTTP (no HTTPS, DNS, etc.)   
  
**Explanation:**   
This policy applies to the backend pod   
It only allows egress (outgoing) traffic to port 80 (HTTP) on the internet   
Any other outbound request (e.g., HTTPS on port 443, or DNS on 53) is blocked   
  
  
Once you apply any NetworkPolicy, it denies all traffic by default unless explicitly allowed — think of it as firewall rules per pod.