**Azure Database for MySQL for AKS Workloads**

* What are the problems with MySQL Pod & Azure Disks?

Ans: In azure disk only one pod can connect to disk at a time. So High Availability concept with single pod is not going to work out.

Drawbacks of Azure Disks –

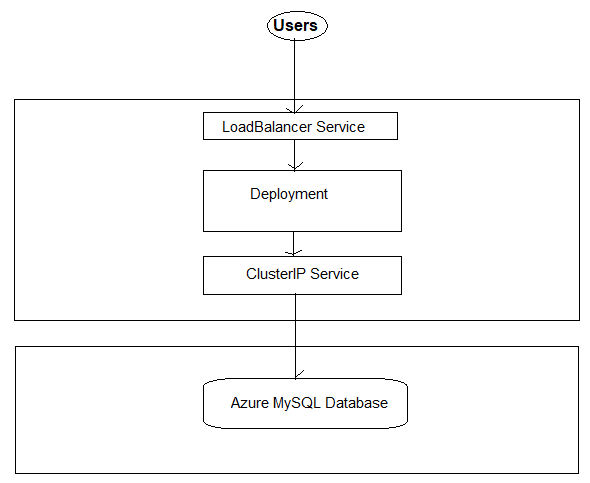
* Complex Setup to achieve HA, Stateful Sets.
* Complex Master-Master MySQL setup.
* Complex Master-Slave MySQL setup.
* No Automatic Backup and Recovery.
* No Auto-Upgrade MySQL.
* How we are going to solve them using Azure MySQL Database?

Ans:

Features:

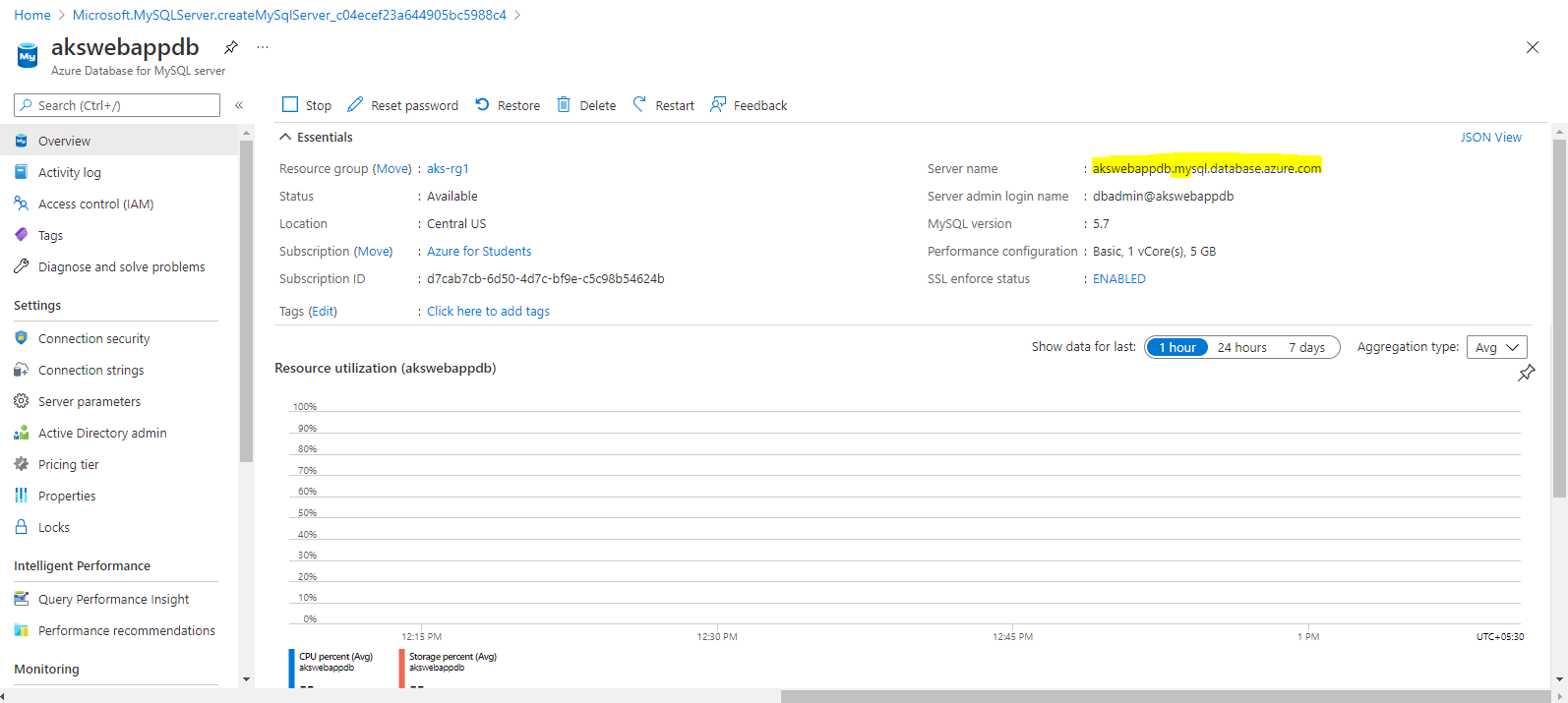
* Built-in HA with no additional cost.
* Predictable performance, using inclusive pay-as-you-go pricing.
* Scale as needed within seconds.
* Secured to protect sensitive data at-rest and in-motion.
* Automatic backups and point-in-time-restore for up to 35 days.
* Enterprise-grade security and compliance.

Diagram



Steps:

1. Create Azure Database for MySQL servers.

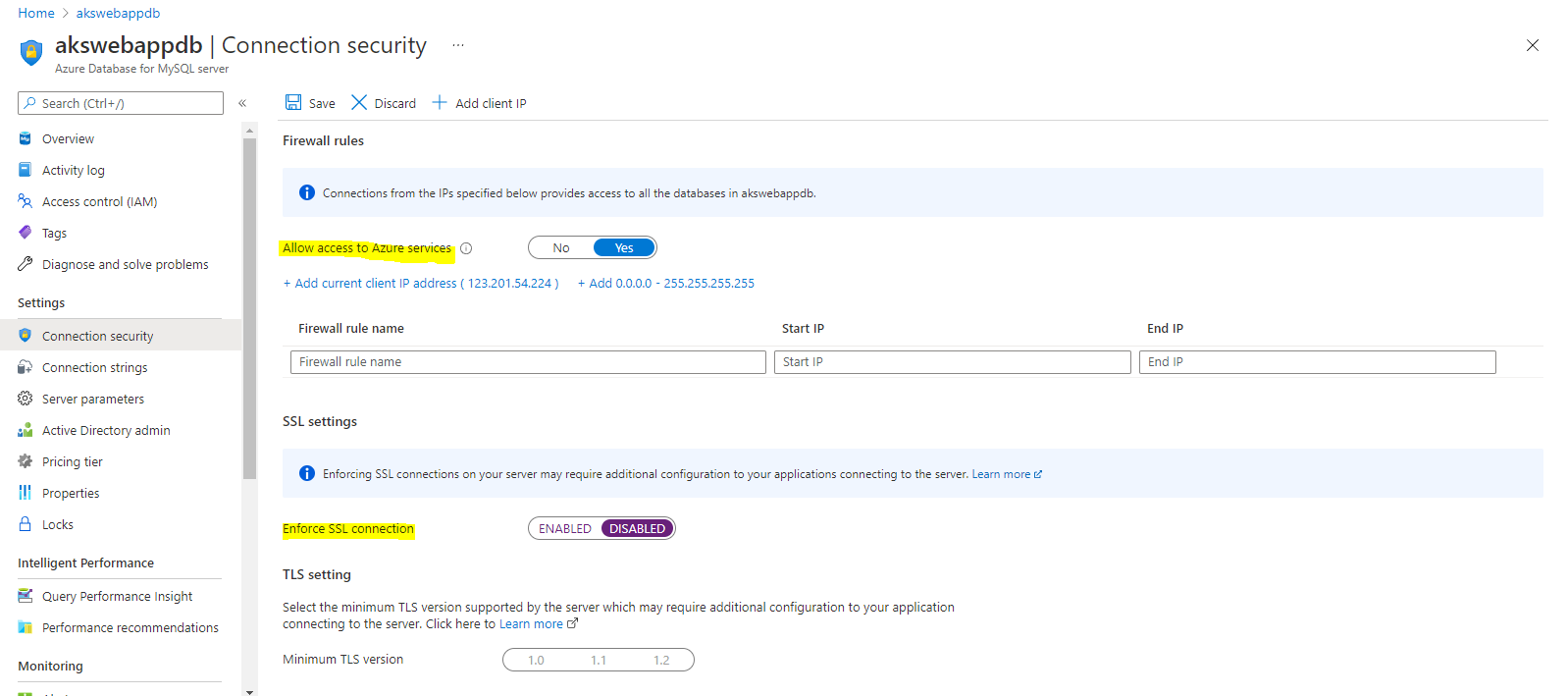


1. Update Security Settings for Database

Go to Azure Database for MySQL Servers -> akswebappdb

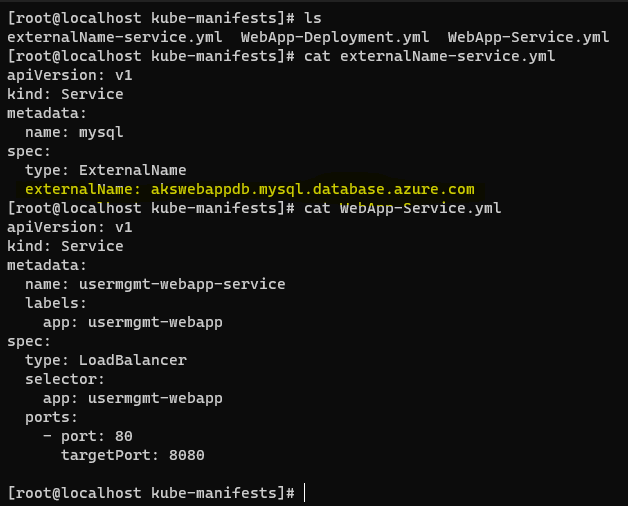
Settings -> Connection Security

* Very Important: Enable Allow Access to Azure Services
* Update Firewall rules to allow from local desktop (Add current client IP Address)
* SSL Settings: Disabled
* Click on Save

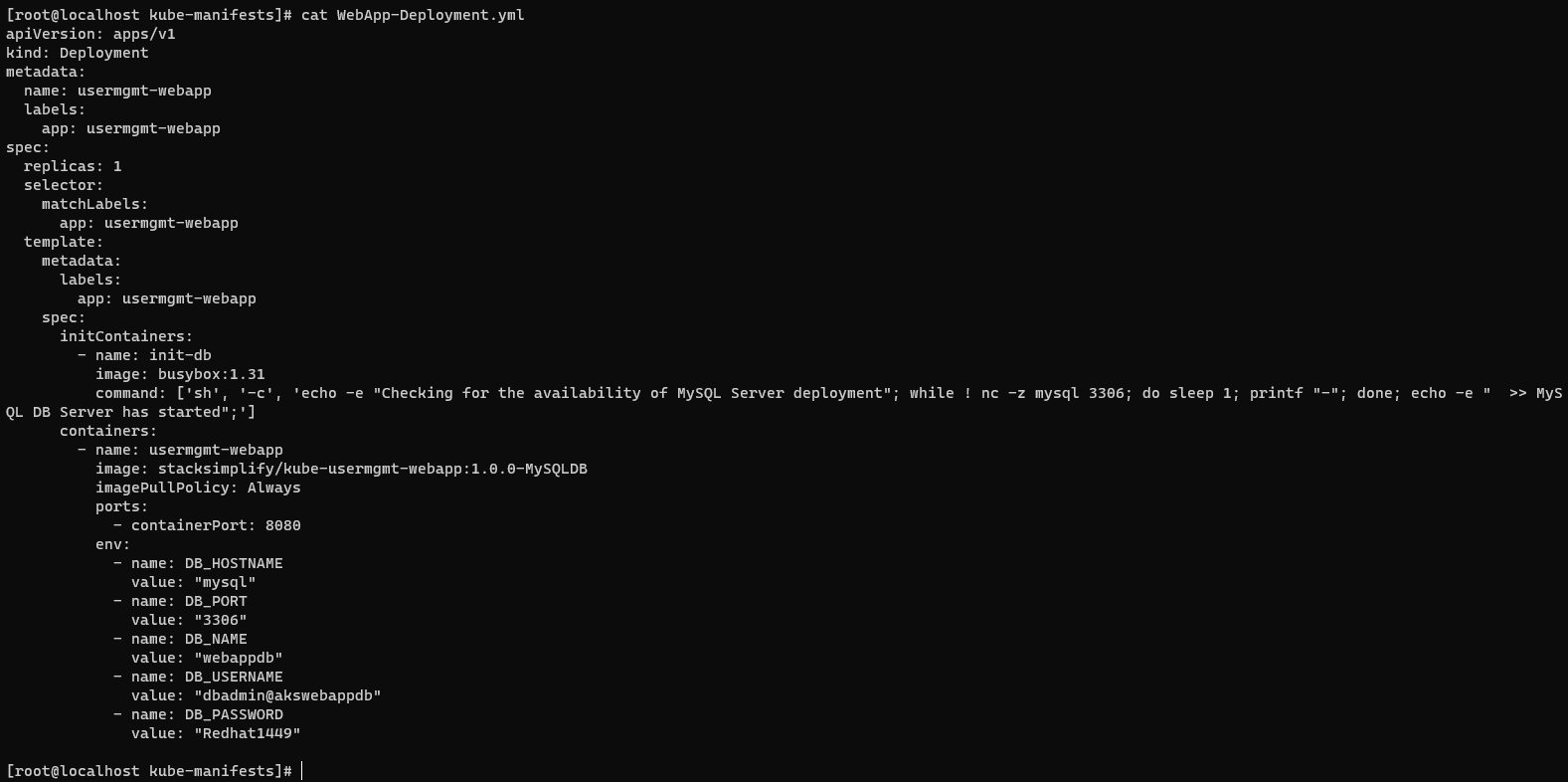


1. Create Kubernetes externalName service Manifest and Deploy

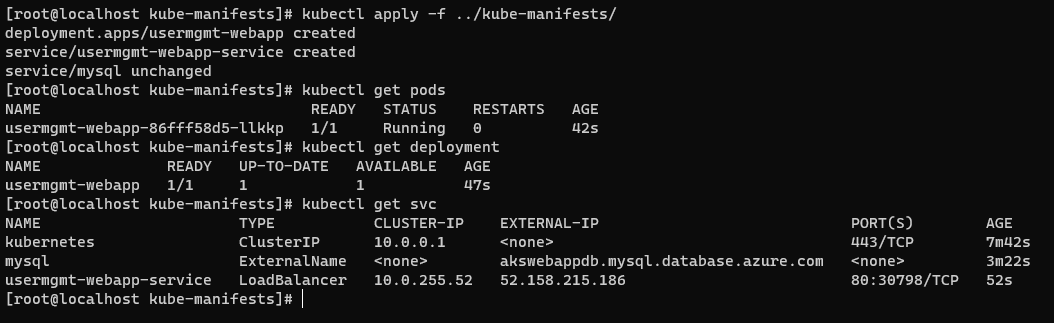
The external service name is MySQL database server name

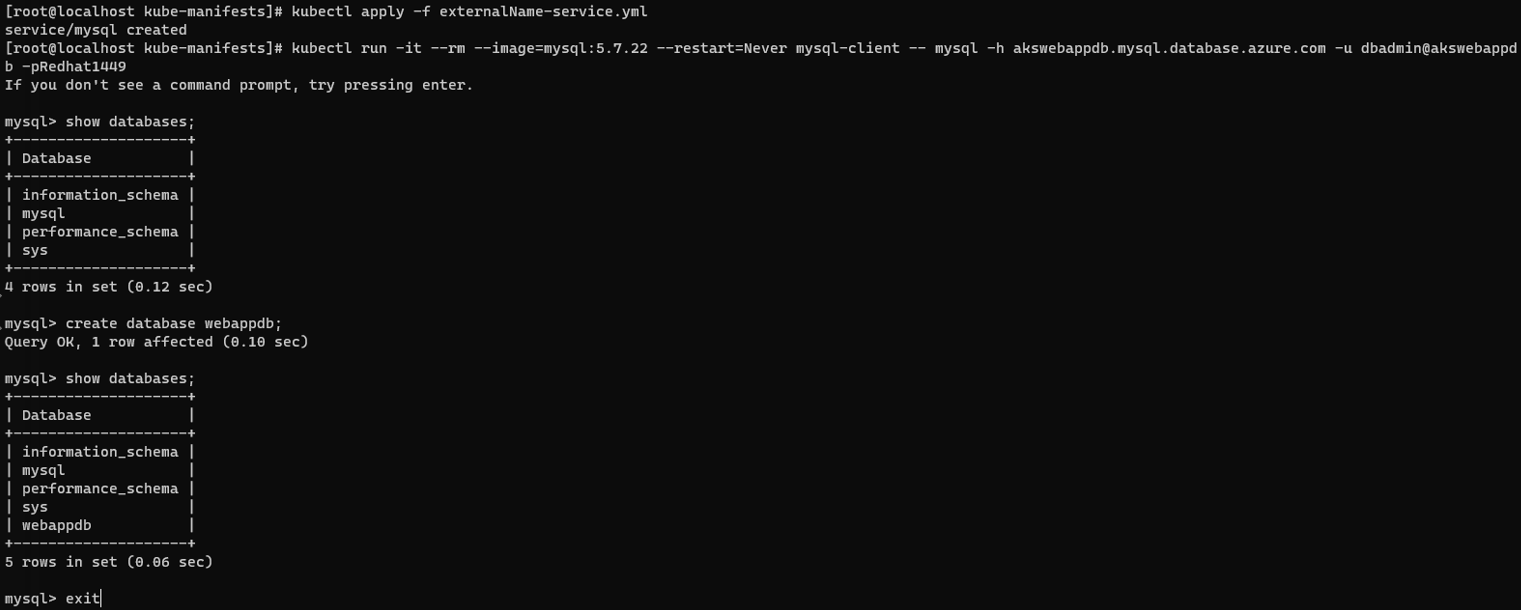


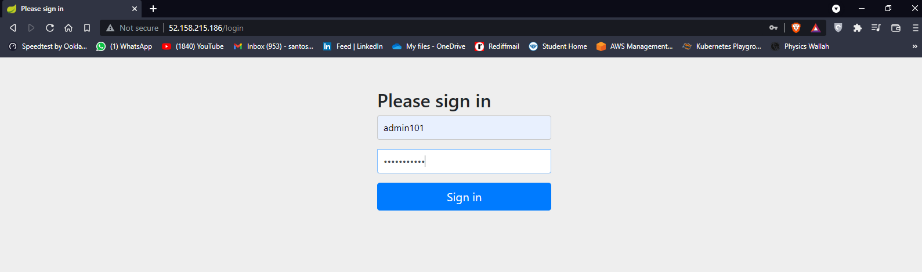
1. Create web App deployment manifest



1. Now deploy everything



1. Connect to MySQL Database and verify the database is created which we provided in ConfigMap.
2. Access Application



Add user and check in MySQL database the user is present or not.  
