# Module 2. App Services

## Topics

* PaaS
* App Services
* App Service Plans
* Web App (Deployment options, Deployment Slots, Alerts, Scale options)
* App Insights
* Kudu console
* Azure Monitor (Metrics, Diagnostics Logs)

## Recommended Courses

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Course Host | Course | $ | Module | MoSCoW | T, h |
| Pluralsight | [Developing with .NET on Microsoft Azure - Getting Started](https://app.pluralsight.com/library/courses/developing-dotnet-microsoft-azure-getting-started/table-of-contents) |  | Building Web Applications and APIs | M | 0.5h |
| Monitoring & Scaling Web Applications and APIs | M | 1h |
| [Design a Monitoring Strategy](https://www.pluralsight.com/courses/microsoft-azure-monitoring-strategy-infrastructure-design)  [for Infrastructure in Microsoft Azure](https://www.pluralsight.com/courses/microsoft-azure-monitoring-strategy-infrastructure-design) | $ | 1. Monitoring Fundamentals | C | 0.5h |
| 2. Using Azure Monitor | C | 0.5h |
| 3. Monitoring Workloads with Application Insights | C | 0.7h |
| 4. Using Log Analytics | C | 1h |
| [Implement Web and Mobile Services on Microsoft Azure](https://www.pluralsight.com/courses/microsoft-azure-implement-web-mobile-services) |  | 3. Deploying and Managing Web Apps | M | 1h |
| 5. Working with API Management | C | 1h |
| 6. Managing Azure App Service API Apps | C | 0.5h |
| 7. Developing Azure App Service Logic Apps | C | 0.7h |
| 8. Creating Azure App Service Mobile Apps | C | 0.5h |
| Edx | [AZURE206x: Microsoft Azure App Service](https://courses.edx.org/courses/course-v1:Microsoft+AZURE206x+2T2018/) |  | 1|Introduction to Azure App Service | S |  |
| 2|Deploying Apps | S |  |
| 3|Scaling and Performance | M |  |
| 6|Monitoring and Troubleshooting Apps | M |  |
| [DEVOPS200.7x: Application Monitoring and Feedback Loops](https://courses.edx.org/courses/course-v1:Microsoft+DEVOPS200.7x+2T2018/course/) |  | 1|Application Monitoring Practices and Principles | S |  |
| 2|Health and Performance Monitoring with Application Insights | S |  |
| 3|Log Analytics | C |  |
| 4|Integrating Third-Party Monitoring Tools | C |  |
| Video Portal | [Microsoft Azure WebApps](https://videoportal.epam.com/video/VolwMmad) |  |  | S | 1h |

## Recommended Reading

* [Web Apps](https://azure.microsoft.com/en-us/services/app-service/web/)
* [Mobile App](https://azure.microsoft.com/en-us/services/app-service/mobile/)
* [API Apps](https://azure.microsoft.com/en-us/services/app-service/api/)
* [Azure Code Samples](https://azure.microsoft.com/en-us/resources/samples/?sort=0&service=app-service) for App Service

## Module Tasks

### Task 1. Deploy Web App

Your customer wants to develop the infrastructure you built in the previous module, and now he is interested in migration his web applications to Azure Web App.

* Create a new Azure Web App service *hr-console-<your-name>* service, and deploy [adventure-forks](https://github.com/epam-lab/adventure-forks/tree/initial) application using VS Publish feature.
* Create a new Azure Web App *service production-manager-<your-name>* service, and deploy adventure-forks application using [Local Git](https://docs.microsoft.com/en-us/azure/app-service/app-service-deploy-local-git) feature.
* Update firewall settings (1433 port) on *aw-vm-sql* VM to allow web app services to connect to the database instance.
* Configure “Entities” connection strings for web app services and make “Departments” tab work.
* Explore web app *Console* functionality.
* Explore Kudu console functionality in *Advanced Tools*.
* Explore *Alerts,* *Diagnostics logs*, *Log stream, Process Explorer* functionality.

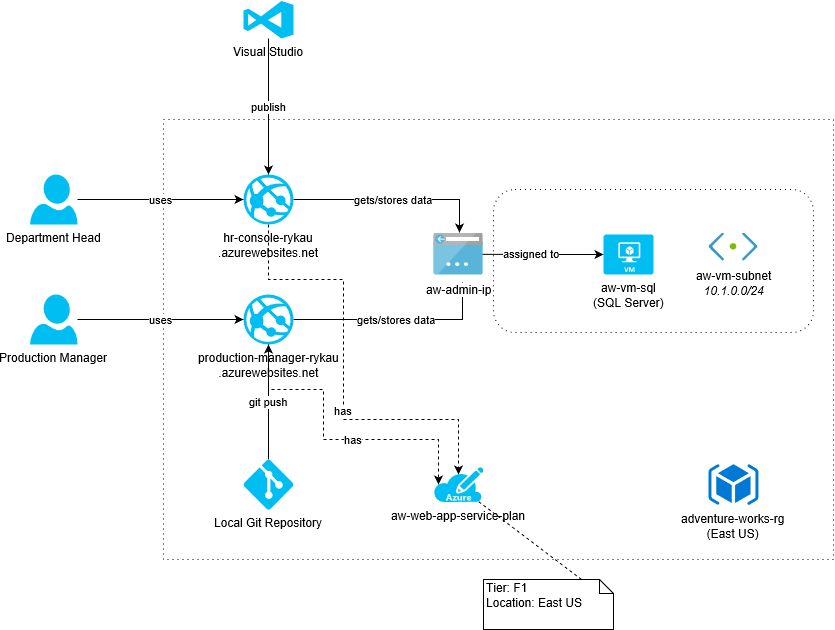


Figure 1. Web Applications connect to the SQL Server using public IP address.

### Task 2. Add App Insights

Enable App Insights for HR Console web application, and add code to the application codebase to track events both on client and server sides:

* /Department page is loaded
* /Department/Employee/{id} page is loaded
* User clicks on /Department/Employee/{id} link

Track events in Azure Monitor.

### Task 3. Create new back-end service

Create a new Web API application with REST API for managing products in Adventure Works database (*Production.Product* table).

* CRUD operations
* Use [Swashbuckle](https://github.com/domaindrivendev/Swashbuckle) for documenting REST API endpoints.
* Use [Serilog](https://serilog.net/) or Log4net for logging exceptions.
* Use public git service (Github or Bitbuck) for storing repository.

Create a new Azure Web App service with S1 production tier, and setup deployment from your git repository to a deployment slot. Explore scaling options.

### Task 4. (Optional) Create advanced infrastructure with virtual network gateway

There is an important security issue in the infrastructure you have built in task #1 – open 1433 port on public IP address. Create a new infrastructure using virtual network gateway that allows Azure Apps to connect to internal networks.

1. Create an Azure CLI script for deploying the infrastructure changes that is shown on the figure 2.
2. Deploy the infrastructure changes (don’t create services that you already have – add new services instead and remove outdated).
3. Export ARM template after deployment.

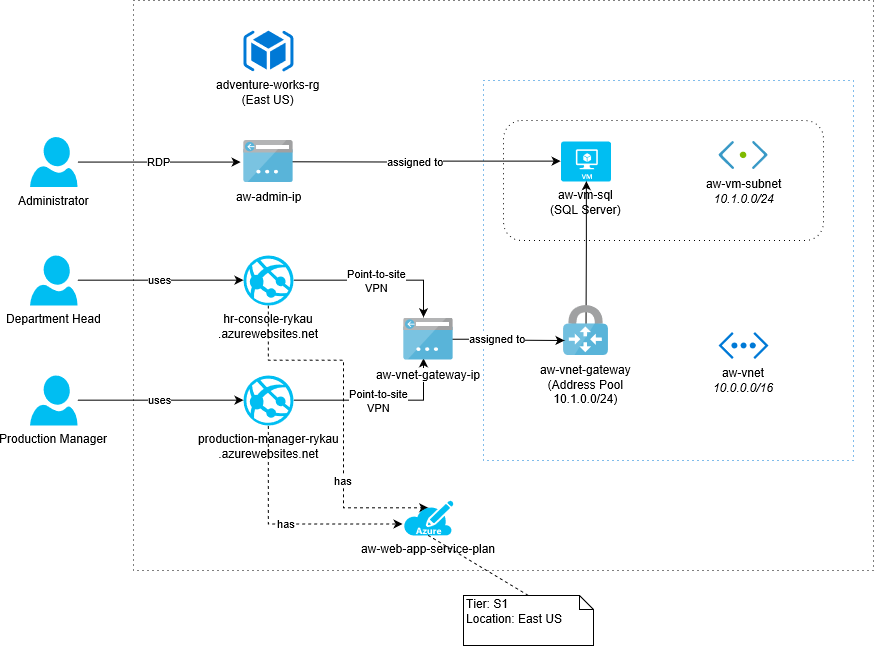


Figure 2. Web Applications connect to the SQL Server using virtual network gateway.