|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Title** | **Work Item Type** | **Effort** | **Tags** | **Description** | **Acceptance Criteria** |
| [21406](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21406) | CH001 Business wants to move to Azure | Feature |  | F001 | GDBC Inc. is a hipster shop that sells old albums on vinyl. They are the number 1 online store that sells these kinds of records. Their business exists for many years and since 1 year they have an online music shop. Last year a new CTO started within the company and announced some big changes in the upcoming years. GDBC Inc. needs to be more efficient since profit margins are suffering. They need to be competitive against the new start-ups who launch idea after idea. Therefore the time to market needs to drastically increase. Because GDBC Inc. is a traditional enterprise company it takes a long time to deliver changes in their software. Last month the CTO announced that GDBC Inc. is going full DevOps. Everything needs to be automated and the target is to release multiple times a day. To facilitate this, the only platform that will be used from now on, is Azure. A cloud platform from Microsoft. The CTO fiercely believes that the move to cloud will improve productivity and agility. |  |
| [21407](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21407) | CH001-AC001 Manually Set up Resource Groups for initial deployment of the web application | Product Backlog Item | 10 | 6K0EDAQE; F001-P001; HELP; REQUIRED | GDBC Inc. wants to move to the cloud. Their data center requires a significant upgrade and the CTO truly believes that the cloud will give GDBC Inc. the flexibility and agility they need to pick up competition. They learned there are alternatives they can use, where they don't need to invest a huge amount of capital, but just pay per use. To get started they want to see what is possible with a simple Proof Of Concept. They just got a trial azure subscription where they now want to deploy the application to see if this is doable. Achievement In this achievement you will use the Azure Team Admin user to create 1 resource group and assign a security group a role to this resource group. This resource group will contain the Web application in the future. | \* One resource group created \* RG-ManualDeploy-Teamname \* Security group assigned Contributor role to the Resource group \* Security group (Teamname) \* Users in the security group should be able to access the Resource Group and be able to create resources (e.g. a Web App) |
| [21408](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21408) | CH001-AC002 Add Azure Web App and SQL Server to the resource group | Product Backlog Item | 10 | 34NIY00O; F001-P002; HELP; REQUIRED | You need to deploy resources to the resource group to make it useful. The resource group is the container where resources can be created that in a sense belong together. The proof of concept that GDBC Inc. wants to create consists of a Web Application and a Database. Achievement In the Azure portal you will create a new Azure Web App with a simple App Service Plan. You will also create an Azure SQL Server that can later be used to contain the data of your Web application. Using the Azure Portal you can connect to the SQL Database to create a table and add some data. Create a table called TestArtist containing an ID and names of artists. Attached you will find a script called create-artists.sql. | \* When you browse to the Azure website, you will see the welcome screen of web apps where they explain how to deploy applications \* A SQL Database has been created containing 1 table TestArtist |
| [21409](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21409) | CH001-AC003 Automate Resource group creation and Security Groups assignment | Product Backlog Item | 15 | F001-P003; XU9BQNU0 | The development team experimented with the manual creation of Resource Groups, but soon found that the manual steps are not the way to go. Everybody needs to be able to create a resource group and this need to be a predictable and reliable process. With the true DevOps Mindset they want to start automating everything. Achievement In this achievement you will create an automation script to automatically create Resource Groups. You will also grant your security group the needed permissions to these groups. This should be done in an automated fashion (Azure CLI / Powershell or...) To get you familiar with Azure and the concept of Resource Groups / Security and Access you need to do the following things. \* Create an automation script to create a resource group for your team \* RG-Playground-Teamname \* Create an automation script to add your AAD Security group to the resource groups \* Security group (Teamname) - Contributor | \* 1 resource groups is created in an automated fashion \* Security Group (Teamname) has Contributor access on the Resource Group. This has been granted in an automated fashion \* Users in the security group should be able to access the Resource Group and be able to create resources (e.g. a Web App) |
| [21410](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21410) | CH001-AC004 Create automation scripts to create Azure resources | Product Backlog Item | 15 | F001-P004; J4JMNKVC; REQUIRED | The team created the required Azure resources to publish their website. But they want to create more environments that are exactly the same so they can use this for testing purposes. One of the benefits of the cloud is they can create new resources fast and discard them when not needed anymore. The team wants to be able to run an automation script that can create the previously created resources (resource group, Web application, SQL server) Achievement In this achievement you will create a Web App and SQL Database in a fully automated fashion. The deployment of the web application is still done from Visual Studio, but at least the creation of the required resources are done automatically. Create a new resource group or use the existing resource groups from previous challenges. | \* A script that can be run from the commandline that creates a Web Application and SQL ServerNote: you should choose a complex password for the SQL Server admin password otherwise it fails setting the password without specific reason \* The Resources are named uniquely based on a provided parameter in the script "Environment\_name". For example when running the script with the parameter "Dev", the Web App is called WebApp-TeamName-Dev. \* After running the script the resources are available in Azure \* The SQL Server firewall settings allows access to Azure services |
| [21411](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21411) | CH001-AC005 Move the automation scripts to Git version control | Product Backlog Item | 5 | CM8UF1K2; F001-P005; REQUIRED | Now that automation scripts are created, the move to DevOps has really started. The automation scripts are available to be used, but we want to keep our sources safe, track the changes we made and be able to use the scripts in our build and release pipeline. This way the team can ensure a reliable and stable way of rolling out new environments on demand. The team wants to use Git as the version control repository and they want to keep the automation separate from the application sources. They create a new Git Repo in the same team project, so the sources are kept separate. Achievement In this achievement you will create a new Git Repository in your VSTS Team project that holds all the automation scripts and Infrastructure as Code files. Call this Git Repo Team-IaC. | \* A new Git Repository that only contains the automation scripts to create the required Azure resources \* Git Repository contains the automation scripts for Resource Group creation and Resources creation \* Team members are able to clone the Git Repo \* Team members can make changes to the Git Repo |
| [21412](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21412) | CH001-AC006 Create ARM templates | Product Backlog Item | 25 | BONUS; F001-P006; UOQPA553 | Automation can be done in many ways. In Azure the templated approach which is also idempotent is by using ARM templates. An ARM template can be deployed using a pipeline or on the command line Achievement In this achievement you will create ARM templates to deploy the Web Application and SQL Server. | \* An ARM template for SQL Server which is triggered from command line and creates SQL Server. \* An ARM template for Web App which is triggered from command line and creates Web App. |
| [21413](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21413) | CH001-AC007 Manually deploy the web application | Product Backlog Item | 10 | F001-P007; REQUIRED; XHSMZZMR | In a previous achievement you have created a Web App and SQL Database in a fully automated fashion. The web application can now be manually deployed from your development environment to the Azure App Service. | \* You have a clone of the GDBC-website repo on your development environment \* The GDBC-website application makes use of the SQL database you have automatically created \* The application builds and runs in your development environment \* The application is manually deployed to Azure from within Visual Studio and runs in Azure. |
| [21414](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21414) | CH002 Remove deployment dependencies from individuals | Feature |  | F002 | After completing the successful Proof Of Concept of running their application in an Azure website, it is now time to make things production ready. The team experienced a problem with the fact that one developer can only publish to the website and that he is not always available. Management does not want to be dependent on one specific person to deploy a new version of the application and asks the team if there is an alternative way to do this. Preferably everyone can release the software to production after they have been asked to do so. The team decides to create a deployment automation that publishes to the Azure website and can be triggered manually. The idea is that the moment a new change is pushed to the Git Repository, the automated build kicks in. After a successful build, a release can be manually triggered that publishes to the production web app in Azure. |  |
| [21415](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21415) | CH002-AC001 Set up a private build agent | Product Backlog Item | 5 | F002-P001; PE0HQ17Z; REQUIRED | In order to build and release the software using the VSTS tooling, you will need a build agent. Next to the hosted agents, provisioned by VSTS, you have the ability to provision your own private build agents. | \* The build agent is installed and configured using the instructions provided by Microsoft. \* Build agent is linked to the already provisioned agent queue for your team. |
| [21416](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21416) | CH002-AC002 Set up a Continuous Integration Build (CI) | Product Backlog Item | 20 | BEB6VJQW; F002-P002; REQUIRED | The GDBC Inc. wants to be able to create the installation package for the website every time a change in source control is pushed. Currently, only one developer can build the software on his own developer machine and this creates a dependency with that person in order to deliver new features. Furthermore the team does not have a validation if the software is still in a deliverable state. | \* When you commit and push new changes to the master branch a build is triggered. \* The build has at least the steps to compile and deliver the WebDeploy package, run the available unit tests and publish this to the artifact store \* After the build is finished you can browse the artifact store and see the zipfile that can be deployed to the webserver. |
| [21417](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21417) | CH002-AC003 Set up a Continuous Deployment (CD) | Product Backlog Item | 20 | F002-P003; LZZ9YC6Q; REQUIRED | GDBC Inc. has a CI build but still has to manually deploy their site to the various test and production environments. They have word documents that can be used as a step by step guide to do the installation, but this has proven to be very error prone. Almost every deployment goes wrong because things are missing. This results in delays or even production incidents. The Management team of GDBC Inc. has asked the team to improve this and the team has proposed to further automate the deployment of the software using a Continuous Deployment Pipeline. To save some costs management has asked to make use on-demand environments | \* When a CI build is finished, the new version of the software is automatically deployed to the Test environment. This test environment is in Azure.Note: a script is provided which you can use to create a Azure Service Principal. This is needed when creating a Azure Resource Manager service endpoint that allows you to connect VSTS to Azure] \* A new environment is created automatically at the beginning of the deployment.Note: a script is provided to create a resource group with a web app] \* The release pipeline contains a way to break down the environment when deployment was done.Note: a script is provided to delete the resource group] |
| [21418](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21418) | CH003 The move to Azure needs to be secure by design | Feature |  | F003 | Things are moving in the right direction. The development teams are picking up and starting to move faster. Azure usage is growing. But .. With great power, comes great responsibility. The Lead Architect raised some concerns about the current Azure situation. He fears that the road to DevOps will eventually cause havoc amongst Development teams. If they can deploy at will, this will cause failures and there is a high risk of failures. To overcome his concerns, the CTO asked him for a solution. This high-level design was the result of this exercise. Developers will get a playground resource group where they can experiment and play. In this resource group, they are owner. When moving to Dev and Test environments they will get access to specific resource groups as Contributor. These subscriptions are perfectly suitable to build and break down dynamically. For Staging and Production, they move to a specific Azure Resource Group. In a later stage this can even be a separate Azure subscription. There the development team are merely readers. To get software and infrastructure deployed into Azure they must use a Visual Studio Team Services Release Pipeline. The connection from the pipeline to the Resource Group is authenticated by a specific Service Principal with elevated privileges. Pipelines need to be used in all environments and need to be there from the start. To keep all secrets out of Source Control and such, Azure Key Vault will be used to keep the secrets. Each resource group will have a dedicated Key Vault for deployment-specific secrets. Team secrets should be stored in a Team Key Vault. |  |
| [21419](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21419) | CH003-AC001 Set up separate Playground Test and Production resource groups | Product Backlog Item | 25 | F003-P001; ZEM3B47P | GDBC Inc. experienced an outage last week, because someone in the team made a mistake with cleaning up resources in Azure. Instead of cleaning the test environment resources, by accident, the production resources got deleted. Luckily the automation scripts and CI/CD pipelines were there to redeploy quite easily, but still, the outage took over one-hour to overcome. GDBC Inc. management wants to get more assurance that this can not happen again and appropriate measures are taken. The team decided to split the various environments also in Azure. For this they want to create separate resource groups and provide access to only the service principal, so creation and clean-up are part of the deployment pipelines. This removes any manual intervention and hence makes the deployments more stable and reliable. | \* There is an automated script to create the 3 resource groups \* One service principal is created that can be used to deploy to all Azure Resource Groups (that the team owns) \* Scripts are committed to the Git Repository so it can be used by the automated deployment pipelines. |
| [21420](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21420) | CH003-AC002 Extend Release pipeline with new environments | Product Backlog Item | 25 | F003-P002; OY608YZQ; REQUIRED | The outage made people aware that is better to roll out gradually before moving to production. The rollout to different environments should be embedded in the pipeline and be triggered every time a change is committed. | \* Release Pipeline creates Azure Resources for Dev, Test and Production \* Release Pipeline contains three environments \* Each environment targets it own Azure resource group \* VSTS Endpoint is configured and used in the Release Pipeline |
| [21421](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21421) | CH003-AC003 Use secrets in your pipeline | Product Backlog Item | 15 | F003-P003; REQUIRED; T1Q0GRH0 | To ensure no secrets end up in Source Control or in plain text, secret variables or a keyvault can be used. Sensitive data that is required for deployment must be inserted in the scripts at deployment time. | \* A SQL Script that creates a SQL user and assigns the user the db\_datareader role \* Script should contain tokens for secrets \* Replacement of secrets with pipeline variables \* Optional; Execution of SQL Script against the created database |
| [21422](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21422) | CH003-AC004 Create a keyvault in your resource groups | Product Backlog Item | 20 | 745TTCXB; F003-P004 | When deploying your software, you most probably have to use some secrets here and there. Think of connection string password, username, and passwords for functional users or SQL users. VSTS has the possibility to store your secrets but once saved this can never be retrieved again. Keyvault has this possibility. | \* Keyvault created in all resource groups \* Service Principal has Get, List permissions on all keyvaults \* Security group has Get, List permissions on Test \* Security group has List permissions on Production \* Random string added as secret in all keyvault with the name [admin-password] \* Release Pipeline uses variable group linked to Keyvault |
| [21423](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21423) | CH003-AC005 Use keyvault secrets in your pipeline | Product Backlog Item | 10 | BONUS; F003-P005; KCURS6C7 | To ensure no secrets end up in Source Control or in plain text, a keyvault is used. Sensitive data that is required for deployment must be inserted in the scripts at deployment time. | \* A variable group is created for all environments which is linked to the keyvaults of the same environments \* Replace the variables with variables from keyvault |
| [21424](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21424) | CH004 Stabilizing the release takes a long time and release is not working | Feature |  | F004 | GDBC Inc. is now very successful in delivering new software, but the team is faced with some nasty production stability issues.Product quality seems to be dropping over time and certain tests are not run, resulting in some frustration in the teams that things got into production while the issues could have been detected beforehand. Management has asked the team to assess the problem an come up with a set of solutions to be implemented in the next sprint. The team found the root cause to be in two primary places: \* Insufficient validation of the software that is committed to the source repository \* A lack of tests that validate some basic scenarios. The team wants to add the tests as standard part of the CI/CD process and add some additional smoke tests. They also want to add code reviews as part of the standard process, not only to prevent bugs to be committed, but also as a means to share knowledge amongst team members. |  |
| [21425](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21425) | CH004-AC001 Run Unit Test in your build | Product Backlog Item | 15 | A5L9JKJZ; F004-P001; REQUIRED | Make sure that your app still works after every check-in and build using VSTS. Find problems earlier by running tests automatically with each build. When your build is done, you can review your test results to start resolving the problems that you find. Achievement In this exercise you will add running unit tests to your VSTS build. A good continuous integration process includes running tests to validate that the code always is in good shape. With code coverage data you can also monitor the trend to ensure that the developers keep adding tests in proportion to added code. \* Configure your build to run unit tests and collect code coverage \* Run a build and inspect the test run information | \* The build definition contains steps to run unit tests and collect code coverage \* A build has successfully been completed with unit test and code coverage metrics \* Test results has been reviewed |
| [21426](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21426) | CH004-AC002 Set up pull requests and short lived feature branches | Product Backlog Item | 10 | F004-P002; REQUIRED; WD7VHGH0 | To ensure the code that is checked into the repository is of the correct quality the team wants to do code reviews. The way to enforce the code reviews is by requiring a pull request before code can be merged into the main branch. A git pull requests has the code review feature as part of the code review. Achievement In this achievement you will use the Pull Request mechanism in VSTS to ensure that every change is reviewed by at least 1 other developer. You will guard the main branch to prevent direct commits by setting up a branch policy. | \* No code can enter the master branch without being reviewed \* Evidence that a code review has been performed when looking at any commit in the master branch |
| [21427](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21427) | CH004-AC003 Create a dynamic provisioned environment to validate the feature branch | Product Backlog Item | 25 | DQXIJ3PG; F004-P003 | To thoroughly test the code on a test environment before the code gets committed to the main branch. You want to test more then only the until tests before you merge to master. Achievement You are going to configure a release so it will also trigger on feature branches. When the release is triggered on a feature branch it will dynamically provision the required azure resources (previous automation scripts can help with this) and deploy our application using the newly provisioned resources. After the release is done, the dynamically provisioned resources must be deleted to prevent unnecessary costs. Note: Resources the provisioned resources should always be deleted even if the release fails or is cancelled else costs will stack up over time anyway! | \* When a changes are pushed to a feature branch, the release is triggered \* When the release runs, your application is deployed on a dynamically provisioned azure environment. |
| [21428](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21428) | CH004-AC004 Run a smoke test as part of the deployment pipeline | Product Backlog Item | 10 | DQDD5M93; F004-P004 | In the past, software got released to production where basic scenarios did not work or the website did not start at all. By adding smoke tests to the deployment pipeline, basic validation is done before moving to the next step. This will prevent releases to go to production when the basics are not working. Achievement In this achievement you will add a smoke test to one of the early stages of the release. The smoke tests only needs to get the HTTP 200 OK response from the main pages of the website. When 200 OK is returned the test is successful. One option is to run Ping command. | \* A smoke test is added to the source repository \* The smoke test is part of the release pipeline |
| [21429](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21429) | CH004-AC005 Run Functional tests in release pipeline | Product Backlog Item | 15 | BONUS; F004-P005; VP9LJY4R | Unit tests are great for validating the logic in the app but we also need functional tests to check that the system behavescorrectly when deployed into a specific environment. With functional tests we test with real data and configuration,test integrations and user interface. Achievement In this challenge you will add functional tests to the deployment pipeline. These are implemented in your solution in the Selenium.Tests or CodedUI.Test projects. If the tests are UI tests interface special configuration of the test agent and the environment is needed. \* Add functional tests to release definition. Functional tests may be both visual and non-visual tests, visual tests need to havean interactive test agent session and the UI infrastructure necessary for the tests to run (i.e. chrome and webdrivers installed). \* Separate configuration from the tests to make it possible to run them in any environment \* Create a release and see how the tests are run as part of the pipeline \* Design a suitable environment for running your tests | \* Functional tests added to release definition \* Code coverage data collected from functional tests \* Functional tests execution as part of the pipeline |
| [21430](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21430) | CH004-AC006 Set the branch policy to automatically validate the pull request with the new build | Product Backlog Item | 10 | BONUS; F004-P006; GT8OCW51 | In order to save time on validation of pull requests we want the pre-checks if the code builds and passes the tests to be done before the pull request is reviewed. Therefore we want VSTS to schedule a build the moment the pull request is filed in the system. Achievement Set the branch policy to automatically run the builds | \* The build is triggered immediatly when a pull request is created. |
| [21431](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21431) | CH005 All software development and deployment needs to be compliant to regulations | Feature |  | F005 | The developers of GDBC Inc. gained quite some speed now they use automated pipelines and cloud technology for their software development and deployments. But just when things started to run smoothly, GDBC Inc. got an audit from the regulatory office around all the financial transactions. The audit report showed some evidence that GDBC Inc. is far from compliant and this should be fixed! Some compliancy experts are hired to look at the process and they wrote a big document on what should be done and changed within the current process. This advice consisted of placing manual approval gates and reviews, creating test plans before every release and test evidence for all the steps that have been performed. The CTO was not amused. All progress that has been made in the development process was going to be obsolete when these rules were followed. Together with some technical experts from various areas they started looking at alternatives. After thoroughly reading all the documents they came up with 4 simple principles that in their opinion was the core of being compliant. \* Ensure integrity by having audit trails on code and artifacts \* Ensure the 4-eyes principle on every change to production To their surprise this was acknowledged by the regulatory office and as long as GDBC Inc. can provide this evidence they are good. The CTO brought everybody together in a room and described some ways to be compliant but still keep their development speed. |  |
| [21432](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21432) | CH005-AC001 All changes to production should be approved | Product Backlog Item | 5 | F005-P001; JI6KK4BY; REQUIRED | GDBC Inc. got an internal audit initiated by its Compliance Officer. His job is to ensure that business operations is in compliance with legislation and regulation. Being compliant reduces the risk of substantial financial or reputational damage to the company. One of the audit issues is that changes directly flow to production. This introduces the risk that code can get in to production that causes damage to the company. There is a decision made to have an approval gate for every change that goes to production. Achievement In this achievement you will set up a manual release gate to approve all the changes. A Pre-Deploy approval is needed from a business user to move a change to production | \* Manual approve of change to production. \* After approval change is deployed. Audit trail is visible in logs. |
| [21433](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21433) | CH005-AC002 Setup branch policies for code review | Product Backlog Item | 10 | CAF5OTKR; F005-P002 | After the audit, it became clear that a developer that makes a change to the code is also the reviewer of that same [code.to](http://code.to/) tackle this the IT department of GDBC Inc. wants to setup a structural code review process. This code review process will not only solve the compliancy audit issue. Since GDBC Inc. is adopting a DevOps way of working the IT department wants to catch bugs or issues as early as possible. Reviewing the code will make catching these bugs or issues shift to the left of the software delivery process. Also the DevOps culture of sharing knowledge will be stimulated, because developers will see each others solutions to a given requirement and educate each other in delivering higher quality code. Achievement In this challenge you will configure a branch policy that will enforce that a code review is required before a pull request can be completed. Setting up a code review process will achieve the following goals: \* Reduce the risk of low quality/malicious/insecure code getting into production that can cause damage to the company in any way \* Minimize the amount of technical debt introduced into a code base \* Increase team collaboration and sharing of knowledge | \* Before a pull request can be completed the code must be reviewed by at least one other developer. \* It is not allowed for a developer to accept his/her own changes \* All reviewers must accept the proposed code change before it can be completed |
| [21434](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21434) | CH005-AC003 Force traceability of work items | Product Backlog Item | 5 | 4Y9I2X0W; F005-P003 | The internal audit also listed a traceability issue. It is not clear which requirement or issue was the reason for the code change.End-to-end traceability was setup, but apparently not all developers make sure the administration to get the audit trail is done. The GDBC Inc. IT department needs to find a way to remember developers about this when they tend to forget and to ensure that traceabilityis always available. Achievement In this challenge you will make sure that a code change is always linked to at least one work item by configuring a branching policy. | \* A pull request must be linked to at least one work item otherwise it will be blocked \* Completing the pull request without a linked workitem fails |
| [21435](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21435) | CH005-AC004 Set up traceability of every change to production | Product Backlog Item | 15 | 3OSBJT6R; F005-P004; REQUIRED | One of the things that needs to be provided as evidence in almost any audit is end-to-end traceability. Who made this change, when was this change made and how did this change end up in the code deployed to production. And is this change really on production and has it not been modified during the deployment process. Achievement In this achievement you will show that compliancy is not as hard as it seems. You will proof to the regulatory office that any requirement can be traced. For that, you are going to implement a change and deploy this to production. \* Create a PBI that describes a change in your backlog \* Write some code (or dummy code) and link this change to the requirement \* Create a build that contains this change \* Run tests in the build \* Create a release that contains this build \* Run tests in the release | \* The release report contains information about the requirement, code changes, tests and builds |
| [21436](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21436) | CH005-AC005 Link test cases to requirements | Product Backlog Item | 15 | 0A6WZO5I; BONUS; F005-P005 | If your test suites include requirements, link these to your test results and view the results on your team's dashboard. This enables end-to-end traceability of requirements for agile teams. For example, when teams do not use planned testing (by creating test plans or test case work items), and instead choose to simply write automated tests that run in the CI/CD pipeline, associating test results with requirements provides an easy way to monitor test results and ensure requirements are met. Achievement In this step you will visualize the quality of developed features using requirement to test association. \* Associate tests with requirements \* Add a Requirements quality widget to the dashboard that shows the test quality for relevant requirements | \* Tests have been associated with requirements \* Requirements with associated tests shown on the dashboard using the Requirements quality widget |
| [21437](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21437) | CH006 Business don't want downtime for the website | Feature |  | F006 | The release pipeline is working out very well for GDBC Inc. and the team is thrilled that they now can roll out changes as soon as they are ready.Unfortunately the business isn't equally happy. Why? Because every time a new feature is deployed the system is unavailable for some time, then it'susually slow for some time and if things don't work as expected the roll-back procedure is time consuming. So this needs to be improved. To minimize the downtime for the website the team has decided to evaluate various techniques that reduce the time the system is unavailable during updates. \* Phased deployments to reduce the risk of deployment by releasing smaller parts at a time (deploy new version of app first, then when stable update the database, or other way around) \* Never break compatibility with existing clients (use telemetry to know when ok to remove feature) \* Add new feature but let old be active (use feature flag to toggle?) \* Make new feature default \* Deprecate old feature \* Monitor use of old feature \* Remove old feature \* Use staging environments (slots) to deploy to a pre-prod environment, which then can be swapped to production \* Use load balancing for scenarios where deployment slots aren't sufficient \* Deploy DB changes incrementally and make sure they are non-breaking |  |
| [21438](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21438) | CH006-AC001 Manually create a deployment slot in your web app and deploy to this | Product Backlog Item | 10 | F006-P001; R7YZ0QQU; REQUIRED | In order to improve the release quality and reduce downtime the team wants to move to deployment slots. The slots help to improve quality by allowing to test the release in a staged environment before it goes into production. When the system is released to production, downtime is reduced by simply swapping the current production environment with the staged environment. Achievement In this challenge you will add deployment slots to the infrastructure to improve the release process. \* Upgrade the app service hosting plan to a tier that supports deployment slots Minimal type "Production S1" \* Add a deployment slot to the web app \* Change the release pipeline to deploy to a slot | \* Web app has deployment slots \* A release has been run that deploys to the deployment slot \* The web app has been tested from the deployment slot |
| [21439](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21439) | CH006-AC002 Automatically create a deployment slot in your web app and deploy to this | Product Backlog Item | 15 | 0EPLD9LA; BONUS; F006-P002 | The manual creation of Deployment slots stresses some of the developers. They don't want to forget this important stuff. In order to make this part of the release process, the creation of deployment slots needs to be automated. Achievement In this challenge you will add a script to automatically create deployment slots to your web application | \* Web app has deployment slots \* A release has been run that deploys to the deployment slot \* The web app has been tested from the deployment slot \* The deployment slot is created as part of the release pipeline |
| [21440](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21440) | CH006-AC003 Update your pipeline to swap slots when ready | Product Backlog Item | 10 | 4Q3DHAPP; F006-P003 | When a system can be deployed to an Azure app service deployment slot it's natural to integrate that with the release managementsystem so that the system first is deployed to a staging environment. The deployment can then be tested in that environment andwhen it is confirmed to be working the staged environment can be swapped with current production version so that the stagebecomes the production system. Achievement In this achievement you will integrate the swapping of Azure web apps from a staged environment (slot) to the production version. \* Add a release phase that swaps the staged environment with the live version \* Add an approval step that waits for the staged environment to be validated before it is swapped to production | \* A release has been run that first deploys to the staged environment, then waits for approval and then swaps the stage to production |
| [21441](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21441) | CH006-AC004 Implement Zero Downtime for Database | Product Backlog Item | 30 | BONUS; D7CIVAVQ; F006-P004 | GDBC Inc sees that the vinyl records are great but there is also a big interest for separate songs (streaming), CD's and merchandise. They decide to create the possibility to also buy these items. Unfortunately the database does not support the desired changes and needs an update that will break existing code. Downtime cannot be afforded, so a plan is created to update the code and database in multiple steps to do a zero-downtime deployment of the database Achievement In this challenge you will implement a strategy for handling zero downtime updates to the database. \* Update the binaries to support multiple database schemas \* Update the database schema with new table \* Migrate existing data to new data structure \* Remove old table \* Change code to support one schema | \* A change to the database has been deployed without breaking existing clients \* A change to the database has been deployed without affecting performance of the running application |
| [21442](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21442) | CH007 Business wants to understand what is used in the application | Feature |  | F007 | Finally the releases are more stable and the speed of new features being implemented is increased. The features that were previously defined to be developed are in progress and business has many ideas on what to improve, implement or change next. However the list is very long, since the speed has been hampered for a long period of time. The team now struggles with what adds most value to build next. They need help in a set of questions that help them set priority. Questions such as: \* What should we prioritize? \* What would have the highest impact? \* Which would help support revenue growth? \* What from the idea lists are still relevant? In the discussions everyone seem to have different ideas and it feels more like politics than acting based on evidence or data. Finally the CTO asked the question - how do we know that implementing these 5 features will support our goals the best? Where is the proof? The data? Instead of guessing, we should understand how our application is being used and what features and how much from our product are being used? Having our product collect such data and provide real live information would be immensely valuable in understanding where the company should invest next. So that is the next highest priority feature to be implemented. We need to know reliably where we should invest next and be able to answer questions such as: \* Who is using our application? \* What actions do they take in the system and what features are not used at all? \* What business flows (multiple actions in sequence that have meaning to business and create value) do they use and are they able to complete the flow or do we have usability problems with current implementations? After some investigations regarding the chosen Azure platform, the team has learned that there is an APM (Application Performance Management) service called Azure Application Insights that can be integrated with the current application to start getting insights into the application usage. the team is asked to do the following: \* Investigate and set up Application Insights in Azure from pipeline \* Investigate and integrate azure application insights into the application to collect usage data \* Implement custom events for every menu item change \* Investigate default Application Insights Usage analysis views and add Azure Dashboard to show usage data |  |
| [21443](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21443) | CH007-AC001 Add App Insights to your WebApp | Product Backlog Item | 10 | 4APQKS0Z; F007-P001; REQUIRED | The GDBC Inc. development team would like to understand their users better and drive business investments and prioritization based on evidence and data instead of guessing. Azure has a Application Performance Monitoring service called Azure Application Insights which allows to gather telemetry to better understand how the business applications are being used. Achievement In this achievement you are going to create application insights resource through Azure Portal and add it to your web application so your team starts receiving telemetry about how the application is being used by users. You are going to enable usage telemetry and integrate that into our web application code. | \* Application Insights resource created in the resource group \* Application code changed to include Application Insights Telemetry |
| [21444](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21444) | CH007-AC002 Set up Application Insights resource in Azure From Pipeline | Product Backlog Item | 20 | F007-P002; XXTW81KZ | Deploying your infrastructure as code in source control, written and runnable in repeatable and desired state manner guarantees that the all environments created are consistently the same, they can always be recreated, would allow validation that the current state of the environment is according to the definition and if its not, the Azure Resource Manager would know what to amend and how to make it so. All resources part of the infrastructure should be deployed in the same manner and Application Insights is merely another azure resource to be deployed. Since GDBC Inc. wants to use application insights to collect data to understand how their application is being used by users, it is important to configure the application insights usage analytics within the application to send information to the newly deployed azure resource The goal is to: \* Have one place where infrastructure is described \* Have one way of deploying infrastructure changes to all environments \* Be able to validate that the infrastructure matches the desired state \* To be able to collect usage analytics to the new application insights resource Achievement GDBC Inc development team already has automation (scripts/ARM template) to create the infrastructure in Azure. Application Insights is a new Azure resource that should be included in the automation so it also gets deployed in the same way. In this achievement you will: \* Change the automation and add the application insight resource to it \* Deploy the new resource using the changed template through the pipeline \* Add usage analytics to pages and configure correct instrumentation key into the application from pipeline | \* Automation that can deploy application insights resource that is configured to be used by the application \* Completed release that has deployed application insights resource to one environment |
| [21445](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21445) | CH007-AC003 Add Application Insights Usage Dashboard | Product Backlog Item | 10 | F007-P003; K81ZHCMF | Now that the application insights resource is being deployed and configured as part of the release pipeline we want to see the usage data that is being collected in application insight. To see the usage data we need to either use the existing Usage Analysis views within application insights or create our own Usages dashboard in azure dashboards. GDBC Inc. business wants to quickly get an overview of relevant usage data from application insights.A great way to bring out only the relevant data and important metrics to business is to create a custom Azure Dashboard that shows just that data and is easy to monitor by the business. Achievement Create a custom azure dashboard that shows users statistics and active sessions / requests / users. Experiment and see what other usages data you can have on dashboard. | \* Custom dashboard has been added to Azure Portal that shows active users / requests / sessions currently and other usage data |
| [21446](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21446) | CH007-AC004 Add Application Insights Custom Events for every Menu Item | Product Backlog Item | 15 | F007-P004; N2YO0SKJ; REQUIRED | The business starts seeing the befenits of having the telemetry in place. Business would like to see additional data on menu items being clicked so they can easily identify what navigation paths and thus features are used. Achievement Track custom event data for every menu item click and make sure its shown on the new Azure Usages Dashboard | \* Menu item clicks are tracked as custom events in application insights \* Custom events are shown on azure usages dashboard |
| [21447](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21447) | CH007-AC005 Add telemetry to measure customer behavior on deleting items in shopping basket | Product Backlog Item | 25 | CBX7UH9W; F007-P005 | GDBC Inc marketing team is constantly striving to improve conversion. the team would like to know if there is a correlation between the fact that people are deleting items from the shopping basket and the total amount of goods they have in the basket. Achievement Therefore they need telemetry data when a user deletes an item from the basket. The moment a customer deletes an item, they want to track the following information: \* Name of the album \* Name of the Artist \* Genre of the album \* Total Amount of the shopping basket Since the amount in the basket can vary a lot, they want that to be bucketed into the following buckets: \* < 100 $ \* between 100 and 200 $ \* between 200 and 500 $ \* over 500 $ The marketing team would like to see the data gathered in a graph and table so they can try to draw conclusions out of the data. In order to get the data, you need to add some C# code to the shopping basket controller. The moment the delete method is called you need to log the additional data using the app insights c# SDK Using the help function will provide guidance on using Application Insights Analytics, yes this comes at a price! | \* On the app insights dashboard there is a graph that shows the genre and amount bucket of deleted records \* On the app insights dashboard there is a table that shows the top 5 genres that get deleted |
| [21448](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21448) | CH008 Optimizing the pipeline | Feature |  | F008 | The GDBC Inc. development team spends a lot of time waiting for confirmations about the state of a deployment during a release, which results in slow cycle times.They want to address this by automating the checks people do during deployment so that the system (VSTS) can detect a good or bad state. If the state is good the systemcan automatically move to the next phase in the release process (typically to update the next environment). The VSTS release process has a concept of approval gates. Gates allow you to query a range of external services, and wait for a positive input from all of them before continuing with a deployment to an environment. When a release is created from a definition that contains gates, the deployment stops until the health signals from all the configured services are successful as illustrated in the image below. |  |
| [21449](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21449) | CH008-AC001 Release only when all requirements are done | Product Backlog Item | 10 | F008-P001; REQUIRED; UT86RB5K | Now that the CI/CD pipeline is working, every check-in from a developer is a potential release candidate. Although it is possible, the Product Owner wants to have some control over what is released to production. He wants to release a set of requirements at once. Work items are nicely tracked in VSTS and he wants to release the software once these items are all set to done Achievement In this achievement you are going to create a Deployment gate based on a Work Item Query. If the query results match your criteria, the release will be automatically approved and deployed. \* Create three (3) work items and tag them with "vNext" \* Create a Work Item query that lists all the "vNext" work items that are NOT done \* Create a pre-deployment gate that approves when this query does not get any results | \* Query for vNext items that are not done \* Query Work Items Deployment gate defined for Production Environment to approve when there are 0 results \* Automatically approved release by Deployment Gate |
| [21450](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21450) | CH008-AC002 Create an Azure Function | Product Backlog Item | 15 | F008-P002; REQUIRED; RT89ZF0W | A situation occurred where there were known issues with some Azure features. The fully automated CI/CD release pipeline, still did a release. Because of this release the application crashed and things got worse. The GDBC Inc. development team want to have the possibility to "stop the line" when there are known issues. They want to address this by creating a check that returns OK or NOK before releasing to production. Achievement In this achievement you will create an Azure function. The purpose of the function is to return status for a GO or NO GO. If the functionreturns OK then all is working as expected and the deployment can continue to the next stage of the deployment. \* Create an Azure function that returns OK or NOK \* Test your Azure function \* (optional) Automate the deployment of the Azure function \* (optional) Automate a test for the Azure function | \* Azure function operational \* Azure function verified in for example Postman |
| [21451](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21451) | CH008-AC003 Add a release gate that triggers the Azure function | Product Backlog Item | 15 | F008-P003; K059DWTH | The Azure Function that returns the status if the deployment can proceed works great. GDBC Inc. wants to make this a part of the automated deployment process so that no manual interaction is needed. Achievement In this achievement you will add a release gate to your release pipeline. The release gate will call an Azure function to decide if the deployment can continue forward. Note: The Azure function to be used here should have been created in an earlier challenge. \* Add a deployment gate to your release definition that calls an Azure function \* Run a new release and see how gates will speed up the release cycle | \* Deployment gate that calls the Azure function added to pipeline \* Completed release that continued deployment based on the gate evaluation |
| [21452](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21452) | CH008-AC004 Add a Release gate on app performance | Product Backlog Item | 15 | F008-P004; F9L88NHP | The last deployment resulted in a panic situation at GDBC Inc. after a few hours the site went down. A thorough root cause analysis showed that a simple code change resulted in a memory leak, which eventually brought down the site. The problem was reproducable in the test environment. The CTO wants that these things are checked before going to production. The Lead Architect knows how to use Azure Monitor Alerts. He wants to use the status of this monitor to decide if the release can proceed. Achievement In this exercise you will create an Azure alert that monitors the behaviour and performance of your application. Next you will add this insight into the release pipeline so that it can continue or halt the release process. \* Create an Azure alert \* Add deployment gate that uses this alert \* Run a new release and validate that the Azure alert is called | \* Azure Alert created \* Azure Alert added to deployment gate \* Completed release that continued deployment based on the gate |
| [21453](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21453) | CH009 More awareness around new feature releases and coolness | Feature |  | F009 | The GDBC Inc. development team are beginning to improve their DevOps processes and they are now releasing more frequently than ever before. However they find that internal teams and stakeholders are not aware of when new features are available, reducing the potential feedback and goodwill. Since they already are using Slack internally, GDBC Inc. want to publish a message for each new internal release to one of the Slack channels. Also, for production releases they want to post a message of the new release to Twitter, to communicate with their customers and end users. When the HR department hears about the social media plans, they directly join in the conversation. In the ongoing war on talent, they want to make GDBC Inc. more cool Any idea is welcome! |  |
| [21454](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21454) | CH009-AC001 Internally - Post to Slack/Teams | Product Backlog Item | 10 | 6D89LPEL; F009-P001 | To get the word out internally about new feature releases, GDBC want to post a message to one of their Slack channels for every new release to the test environment. Achievement In this achievement you will extend your release definition to post a Slack message after a successful deployment to the test environment. \* GDBC is cannot provide a Slack Workspace, use an existing (at your own risk!) or create a new Slack Workspace | \* A message in a Slack channel or MS Teams channel (use your own or create one) for every successful deployment to the test environment |
| [21455](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21455) | CH009-AC002 Externally - Post to Twitter.. With some GDBC details | Product Backlog Item | 20 | F009-P002; REQUIRED; VPAQUBY1 | Every new release to production is worthy of some publicity. GDBC. Inc wants to create some buzz by publishing a message to Twitter for every new release to production. Achievement In this achievement you will extend your release definition to send a tweet every time a new deployment has made it all the way to production. | \* A tweet with the hash tags #gdbc and #releasetweet is sent when a new release has been deployed into production |
| [21456](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21456) | CH009-AC003 Show some DevOps Coolness by posting a DevOps Team selfie | Product Backlog Item | 30 | BONUS; F009-P003; MEXFKO9U; REQUIRED | As an answer to the request of the HR department, one of the developers came with great idea. Post a selfie of your DevOps Team approving a Pull Request, or any other cool action they do every day. The HR hopes that this contributes in the coolness of the company. Showing the fun and coolness of the products they use on a day to day bases Achievement You will create a selfie of you or your team doing some cool stuff. Post the selfie on Twitter (required), Facebook,Instagram or Linked (or all) and tag them with #GDBC #selfie and mention @gdevopsbc (on twitter) | \* Selfie tweet/post sent |
| [21457](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21457) | CH010 Business want to control releases of a feature | Feature |  | F010 | At the launch party of a well known artist GDBC Inc. wanted to make a big impression. They made a unique deal with this artist to have the exclusive right to sell the music for at least a month. The idea was that the landing page of the web application showed this news so people would directly be attracted to this exclusive offer. The development team worked day and night to get the web application ready and just before the press conference started they released their web app. Unfortunately, that did not go well. The release failed, and the press conference could only show screenshots. After a very well organized blameless post-mortem it became clear that this error was caused by the fact that there was no real time to test. GDBC Inc. decided to start looking at ways how the business could be enabled to do their releases themselves. Separating the functional release form the technical deployment. This way, they can decide when something goes live without being dependent on a technical release that can fail. |  |
| [21458](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21458) | CH010-AC001 Hide a discount banner and sales funnel with a feature toggle | Product Backlog Item | 10 | DMZX0J8X; F010-P001; REQUIRED | The marketing and sales department of GDBC Inc. are working on a global marketing campaign during which a few top selling products will be offered with significant price reductionsduring a limited time window. The exact date at which the campaign will be broadcasted on television isn't known yet. It is important that the discount banner will be displayed on thewebsite together with or shortly after the television broadcast. The goal is to: \* Get feedback on functionality that has been activated \* Decouple deployment and exposure of new functionality Achievement In this achievement you will prepare the application for the upcoming marketing campaign. \* Add code to the application that is required to support feature flags \* Test if the banner is displayed based on the feature flag setting. | \* The discount text/banner on the homepage can be (de)activated with a feature flag \* The feature flag setting can be set through configuration |
| [21459](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21459) | CH010-AC002 Create a feature toggle switch page to (de)activate a feature | Product Backlog Item | 20 | F010-P002; IFS77399 | The campaign was launched, but some unexpected errors occurred. Solving those errors took a lot of time and the lead-time for thesechanges to be solved in production environment was quit long. In the mean time errors kept occurring until someone of the IT staff, who wherenot available in the weekend the campaign launched and during the week where all intensively working on problem solving, had some time to changethe web application configuration.For the future, the marketing and sales department of GDBC Inc. wants to be able to quickly (de)activate the functionality when required.Because of this they want an admin user to be able to (de)activate the feature flag through the administration page of the web application. The goal is to: \* Turn off new functionality quickly in case something goes wrong and problem solving takes a long time. \* Make these changes without redeployment of the web application Achievement In this achievement you will prepare the application so that an admin user can (de)activate the campaign functionalityfor the upcoming marketing campaign. \* Add the option (de)activate the discount and sales functionality through the administration page replacing (de)activation through web.config \* Change the code that uses the web.config value to using the value set through the administration page \* Test if the feature is (de)activated when the setting is changed on the administration page | \* The feature flag setting can be set through the administration page by an admin user instead of changing the application configuration |
| [21460](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21460) | CH010-AC003 Configure the release pipeline to gradually roll out the application | Product Backlog Item | 30 | 7584J270; BONUS; F010-P003 | This first marketing campaign was a huge success for GDBC Inc., but the journey had some hiccups. The goal is to: \* Get feedback early \* Limit the impact if there is an error in brand new functionality or to test if customers like newly added functionality. Achievement In this achievement you will setup the release pipeline and the app service in such a way that new functionality is exposedto an incrementally growing group of users. In other words your are applying the concept of a canary release. \* Deploy the web application to a deployment slot and route different percentages of traffic to this application by using the Testing In Production (TIP) feature of an App Service. \* Implement deployment rings in your release pipeline and automatically increase the percentage of users after approval has been given for each ring \* After successfully passing your approval stages swap the deployment slot and send all traffic to it | \* There are one or more rings in the release pipeline to which you the application is deployed after approval has been given. \* For each ring the amount of traffic that is directed to this ring increases. \* In the final ring all traffic is directed to the newly deployed application. |
| [21461](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21461) | CH011 Business hears complaints about performance during busy times | Feature |  | F011 | The GDBC Inc. development team is in trouble, either something has impacted performance in the last few deployments or there has been a lot of behaviour change. In any case, there is a significant performance degradation in the last weeks and its especially visible during the busy hours. The first line support has been getting increasing number of calls from unhappy and frustrated users. Unfortunately the support cannot answer all the questions because they have a hard time finding the causes. After a constructive meeting the Development team came up with a few good questions. \* How to find out what is going on and in the future know if the latest deployed application is the cause of performance degradation? \* How can we be proactive and know that there is performance degradation happening without getting angry calls from users? \* How to get more data to investigate what is causing the performance degradation and how many users are impacted by it? One of the developers remembers that they integrated Azure Application Insights into the application for usage analytics, but its a full Application Performance Management Service and can also help with getting control over the performance issues they are facing and answer all the above questions \* Set up load tests to monitor site availability to be able to be proactive about performance issues \* Integrate Application Insights to gain full insights into application performance from front to back \* Set up release annotation in the pipeline to create a marker in application insights so you can correlate performance and other data with deployments \* Add performance metrics to azure dashboard to monitor |  |
| [21462](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21462) | CH011-AC001 Add Application Insights to catch performance and diagnostics | Product Backlog Item | 20 | F011-P001; REQUIRED; T5QFCD73 | The GDBC Inc. development team has trouble figuring out what is going wrong with the application and especially be proactive about this. The information regarding all the different application components and dependencies would be super helpful in these scenarios since there's hardly any exceptions until things go horribly bad. The Development team is quite sure that if they would have telemetry around resource utilization, communication between components and memory usage, they might be able to mitigate these issues much faster. Achievement In this achievement you are going to use Application Insights throughout all application components and layers by integrating it through the code. By using the necessary NuGet packages within the application the telemetry can be send to Application Insights. | \* Web application Live Metrics can be viewed from the Azure Portal \* CPU and Memory can be viewed from the Azure Portal \* Diagnostics and Exceptions show up in the Azure Portal \* Use Application Insights Snapshot Debugger to analyse application run-time state |
| [21463](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21463) | CH011-AC002 Add Release Annotations to your pipeline to show when a release has been done | Product Backlog Item | 10 | F011-P002; JYV380P0 | The GDBC Inc. development team feels pretty much on top of performance and application troubleshooting since they know when performance issues occur and they have ways to diagnose and investigate them all because of collecting necessary telemetry regarding the running application components and dependencies. It feels pretty good to be proactive.Lately though some of the performance issues seem to be correlated always with releases, but no-one is quite sure. Deployments happen quite often now and its hard to know if the latest deployment could have affected the performance or not. Both the CTO and Product Management is eager to get some clarity if the fast and frequent deployments could be the culprit and in those cases extra performance tests and quality gates / verification should be implemented to have more stable versions of software. Application Insights has a feature called Release Annotations. These are markers in application insights regarding important milestones or changes that happened to resources (application components) that send telemetry. These markers are visible on performance and metrics dashboards thus allowing to correlate change in metric with an important change such as new version deployment. Achievement Add release annotation creation to your release pipeline so that there is a marker in application insights for every new version that gets deployed. | \* Release annotation creation step in the pipeline \* A release has been run that has created a marker in application insights |
| [21464](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21464) | CH011-AC003 Set up a availability and load test for the application | Product Backlog Item | 15 | BONUS; F011-P003; TU78PU07 | Some users report some pages that are not available on specific times. After some investigation this had various reasons. To be more proactive in mitigating problems before they are deployed to production, the Lead Architect wants to make sure that the most important parts of the application are checked on a regular bases. He also want to hit the application with some load to see if this changes behaviour. Application Insights has a feature called "availability" that offers the functionality to check the availability of your application by running a test on those pages on a regular interval from multiple regions. These web tests can be either simple ping tests from multiple locations or multi-step tests that go through a sequence of business functionality (such as going to home page, logging in, navigating to x). Achievement Setup single step availability test called URL ping test that would ping your application and continuously monitor the availability and responsiveness. Set it up with 5 minute interval and from 4 locations that are relevant to GDBC Inc application (your choice). | \* Single url ping test setup and running / monitoring availability and responsiveness \* Test trial application unavailability to confirm good and bad scenario visibility \* Add site availability to your azure dashboard |
| [21465](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21465) | CH011-AC004 Add Performance Metrics to Azure Dashboard and alert on VSTS dashboard | Product Backlog Item | 15 | 7GMAE1P6; BONUS; F011-P004 | GDBC Inc. development team has gathered a ton of metrics, but the information is scattered in different dashboards within application insights.The CTO would like to get a more high level and informative dashboard regarding the current and last 7 days performance metrics of the application to use these to monitor the application. This will be useful for everyone - the support, live site / operations and development team and of course the CTO that will most likely take nice snapshots out of them to illustrate and explain situations to management. Achievement Create an azure dashboard that would pull together different performance metrics that are now collected regarding the application and its components and dependencies and offer high level and detailed view over those metrics. The dashboard should cover: \* Application components / dependencies health state \* Response times \* Availability and responsiveness information \* Users / Requests / Sessions \* Resource consumption and availability (memory, cpu, io, etc. of relevant and important resources) \* Performance counters \* Exceptions / sec rate | \* Created a dashboard with available and relevant metrics and views on them to provide high level and detail information \* Shared the dashboard within organization by adding metrics from Application Insights Analytics to VSTS dashboard |
| [21466](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21466) | CH012 Developers complain about long time in setting up their Dev Environment | Feature |  | F012 | The GDBC Inc. development team spends a lot of time setting up their local development environments. They need to install the correct tools, such as SQL Server, and also get the application in the correct state in order to start making changes to it and be productive. Using Docker containers can make development teams a lot more productive, since they allow them to define everything that an application needs to build and run in a simple text file that can be checked in together with the rest of the source code. |  |
| [21467](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21467) | CH012-AC001 Create a container for WebApp | Product Backlog Item | 15 | F012-P001; RLUIVMXP | By containerizing the web application, developers will be able to develop and run the application locally in a production-like environment, without having to install anything else than Docker for Windows on their local machine. It will also improve the application delivery since the same image will be used throughout all environments, including the production environment. Achievement In this challenge you will add Docker support to the existing web application and verify that you can debug and run the application locally inside a container. Note: The web application will not be fully functional yet. If you see the yellow screen of death running in your container, then the mission of this challenge is still accomplished, since we will add the SQL Server in a subsequent challenge. | \* Docker support added, with a docker compose project and a Dockerfile for the web project \* When the project is started, the web application should be running inside a Docker container |
| [21468](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21468) | CH012-AC002 Set up a SQL Server container | Product Backlog Item | 15 | F012-P002; OB9OG2DI; REQUIRED | Getting the database in a correct state, with the right version of the schema and the necessary metadata is something that many teams struggle with. Containers can help out a lot here, by packaging the necessary version of the SQL server and Achievement In this challenge you will add a SQL Server container to the solution, making it easy to develop locally against the correct version of the SQL Server. | \* When the application is started, there should be two containers running. One with the web application, the other one with SQL Server \* The application should be working, running against the database in the other container \* Successfully connect to the running instance of SQL Server inside the container using SQL Management studio |
| [21469](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21469) | CH012-AC003 Create Prefilled Data in SQL Server container | Product Backlog Item | 15 | F012-P003; REQUIRED; TPEI9RZI | Creating a database with prefilled data by running migrations scripts with data can be time consuming, especially if we want to use it as a basis for automated unit testing. Having a prebuild container image with SQL Server and a database with all necessary data available for developers and testers can improve the overall delivery process. Achievement In this challenge you will use the running SQL container from the previous exercise as the basis for a new image. In the next challenge, you will push this image to ACR, together with the application image. You will also change the application so that is does not create a database when starting up. | \* A container image with the MusicStore database \* MvcMusic store application should use this image instead of the default SQL Server image |
| [21470](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21470) | CH012-AC004 Set up Azure Container Registry | Product Backlog Item | 10 | 25IBATYF; F012-P004; REQUIRED | When working with Docker containers, the artefacts that are produced are Docker images, that contains the application as well as everything needed to run that application, including the base OS. These images need to be published to a container registry, from which they can be pulled from different target environments and started as containers. Achievement In this challenge you will create an Azure Container Registry that you can use later on to publish the images of the application | \* A generated Azure Container Registry visible in the Azure portal |
| [21471](https://globaldevopsbootcamp-emea.visualstudio.com/97b42831-e7dc-400b-9012-9261d9fb1478/_workitems/edit/21471) | CH012-AC005 Publish "development" Container to ACR from pipeline | Product Backlog Item | 15 | F012-P005; J677O2SM | Even if we are not running the application using containers in production, they can still provide a lot of value for the development workflow. We can for example publish development images of the application that developers can pull and run on their local machine, without having to compile and configure them. Achievement In this challenge you publish container images that you have built locally to the Azure Container Registry that you created previously | \* At least one container image that has been pushed to the container registry |