Devops Portfolio

# MICHAEL PLASOM-SCOTT

For DWP Digital and Makers Academy

## Contents

##### About

[Intro To Me](#_Intro_To_Me)

[The Team](#_The_Team)

[My Role](#_My_Role)

[Tech Stack](#_Tech_Stack)

##### Portfolio

[Ticket Name](#_Ticket_Name)

[Ticket Description](#_Ticket_Description)

[Ticket Process](#_Ticket_Details)

[Knowledge](#_Knowledge)

[Skill](#_Skill)

[Behaviour](#_Behaviour)

##### DUTIES

[Knowledge](#_Knowledge)

[Skill](#_Skill)

[Behaviour](#_Behaviour)

## About

##### Intro To Me

I'm Michael Plasom-Scott. I studied as a Graphic Designer, worked as a teaching assistant and technician in a school, and I'm now re-training as a DevOps Engineer. I come from a creative and collaborative background, that although does not scream DevOps has been vital to my development in the role.

This course is a brilliant opportunity to have a glimpse into a fascinating industry, completely unlike anything I've tackled before. Every day is a new challenge, and I haven't stopped learning since day one.

##### The Team

My team is an agile team led by Tom Farrow. Along with my Line Manager, Shazad Azam, they have been extremely helpful in helping me get to grips with the day-to-day life and work ethic of a DevOps engineer. The team has daily morning standups in which we talk about what was achieved yesterday, what we plan to achieve in the coming day, and any blockers that might stand in our way. It’s also an opportunity to ask any questions or for advice. Tickets are assigned on an ad hoc basis, and completed as quickly as possible, with as much communication as possible. I spent a lot of my time at the start of my placement shadowing Luis Castromil, another DevOps engineer, who moved on to giving me tickets to try myself.

We work alongside various other teams with different organisational cultures. Our team of DevOps engineers runs very much on an ad hoc basis, picking up work as and when it needs to be done. This works well for us and enables us to be flexible and efficient in assigning work every morning and letting team members support projects to completion. Because of the ad hoc allocation of tickets, our backlog of tickets and technical debt is often managed inefficiently and needs to be addressed in occasional calls where we divvy up some backlog tasks. One of the teams we work with is the product team, who track tickets and changes closely using tools like Jira and Confluence, and work in sprints. They use a regimented Scrum framework, utilising particular ceremonies to keep on top of their work. As a team they work efficiently with this level of governance, though because they work in sprints, if we need to change something we must wait for their next scheduled sprint, meaning some of our tasks can be delayed. Although these two methods of working – one more ad hoc and one more regimented – are both efficient for the teams in question, they can sometimes clash when we need something from each other.  
[K9](#K9)

##### My Role

My role is that of Apprentice DevOps Engineer in the Digital Shared Channels Experience (DSCE) at DWP. I often work within the Notifications team, however some projects I do affect multiple teams within the DSCE Directorate.

##### Tech Stack

AWS

S3

Lambda

Cloudwatch

DynamoDB

Terraform

Ansible

Prometheus

Grafana

Gitlab

## Portfolio

Project #1

##### Ticket Name

NSE-2264

##### Ticket Description

Create an S3 Bucket with the local name Data Migration Changes. S3 access then provided to Data Specialist who will pull the files Users.CSV and Org\_Node.CSV that will be uploaded to SharePoint. Users.CSV needs to be encrypted. Files to be exported to S3 location Wed/Fri by 8am.

This ticket was assigned to me as a way of getting me used to an agile workflow and working alongside developers to roll out some changes.Graphical user interface, text, application, email

Description automatically generated

Figure 1 - Screenshot of ticket with full technical details of resources needed

##### Ticket process

Meeting with Luis  
The process started with a call with Luis where he outlined what would need to be done, and then updated the ticket with a bullet pointed list of everything. He also said one of the first things to tackle would be the creation of a GitLab CI/CD pipeline. Tyler and I said we’d split up the basic work, he’d focus on writing the terraform of the Lambda and I’d focus on the pipeline.   
[S8](#Skill_1)

Creating the pipeline  
I spent a while on the initial stage of the pipeline. It was my first time creating a GitLab CI/CD pipeline at DWP, so I was unaware of the protocol used. I looked at other examples of pipelines from different repositories that the team had worked on and spent a while trying to decipher how they worked. Working on my own branch in the repository, I pushed up a few very simple versions of what I understood, to try and see if I could get something minimal off the ground and build it up (fig. 2). Table

Description automatically generated

Figure 2 - Testing scripts in a barebones pipeline (failing)

I couldn’t understand why these were failing so I set up a call with Luis and we workshopped around the setup of a pipeline. He explained about the use of fragments, and the ‘extends’ function in a Gitlab CI file.

Text

Description automatically generated with medium confidence

Figure 3 - After workshopping, part of the pipeline file

We did some productive pair programming, building up a pipeline that would pass correctly. Initially Luis used a Tour Guide style of pair programming, explaining the fundamentals of the pipeline, and what each component did. Then we used a Driver/Navigator style, switching between me sharing my screen and taking directions from him, and me making suggestions while he screenshared. I asked if Ping Pong style pairing might be suitable for this, but as we weren’t formally writing tests, we stuck to switching between who was driving and who was navigating in a style similar to Ping Pong, but without the formality of TDD. After that it became a lot clearer and I eventually pushed up a merge request with a pipeline that worked as it should, with the correct steps in it (fig.3).   
[S13](#Skill_1), [K20](#Knowledge_1)

*Chart

Description automatically generated with low confidence*

Figure 4 - The earliest MR is the one on which Luis and I were doing lots of testing. It is failing at the Unit Test stage only, so after the developers do the unit testing the pipeline passes happily.

Code review by Luis  
Once the pipeline was working as expected, Luis took the time to review the code that Tyler and I had in the repo. This was an incredibly helpful step, and one of the key parts of a smooth and agile DevOps environment. Often it was small things like not following the correct naming conventions (fig. 1), however Luis would often take the time to highlight important DevOps principles, as a learning opportunity for me (fig. 2).   
[B1](#Behaviour_1)Graphical user interface, text, application, email

Description automatically generated

Figure 5 - Code Review comment from Luis

Graphical user interface, text, application

Description automatically generated

Figure 6 - Code Review comment highlighting DevOps Principle

Fixing Changes  
Following the code review I worked by myself to make sure the suitable changes were applied. This involved a lot of simple changes to variables within the file, which were simple. However, there were some elements that I was unfamiliar with, particularly the scheduling of Lambda functions, as was required by the ticket - Files to be exported to S3 location Wed/Fri by 8am - so I researched around ways to do this and looked in some other repos from the team to find examples of cron triggers. I settled on setting up the cron expression (fig. 3), so the file was exported on Wednesday and Friday at 7:30 and ran this by one of the [Business Analysts/Architects] to make sure that was a suitable time.   
[S8](#Skill_1)

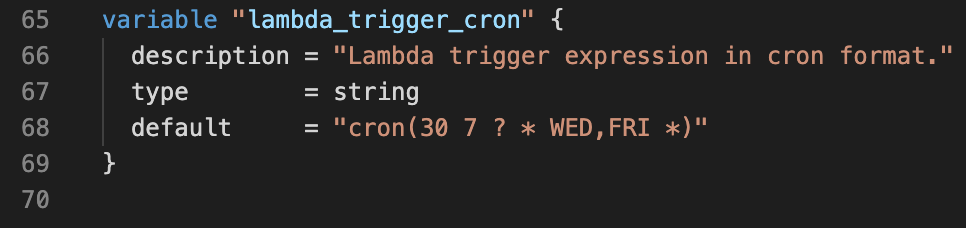
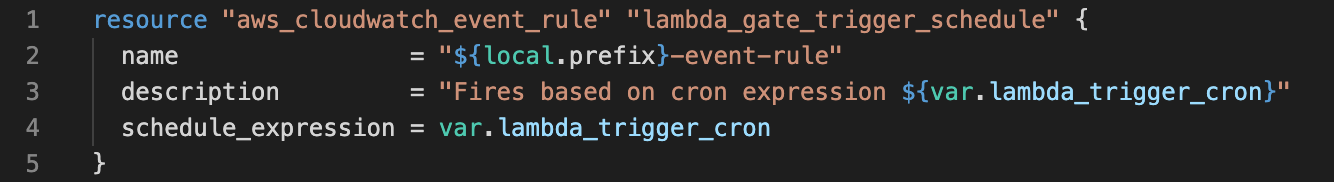


Figure 7 - Cron Trigger

Debugging with Developers  
Once the setup was all complete (in theory) from the DevOps side, a chat was created with me, Luis, Daniel, and Ritesh. Daniel and Ritesh were the developers adding some code in, using the pipeline I’d created. At this point some of the steps in the CI/CD pipeline started to fail – mainly catching errors in the code and doing what the pipeline is supposed to do in part, i.e. catch errors before going to deployment. However, one of the times it failed was due to an error in the Terraform code – the handler path in the Lambda config had been copied across from another repository and was incorrect, so I updated the code and pushed again. Another error that slipped through the net was that the Lambda wasn’t given enough time to connect to the database, having 3 seconds as a default. This gave the Devs an error message, which they fed back to me, and I fixed in the Terraform code (fig. 8). This process is greatly enhanced by the emphasis DWP put on a blameless culture i.e. errors in code are not seen as someone’s “fault”, rather just as a learning opportunity for everyone. Constant fixes and improvements are required and facilitated by everyone, without finger pointing. This improves efficiency, and makes everything both more smooth and more versatile for future use.  
[K23](#Knowledge_1)

Graphical user interface, text

Description automatically generated

Figure 8 – Devs highlighting error

Deploying to DSE-1  
Having fixed the bugs and got the pipeline running smoothly, we finished for the weekend. On Monday Luis started a call with me and demonstrated to me the way that we deploy code into the DSE-1 (Develop Supported) environment.

Business Impact  
The changes we made created an AWS Lambda that retrieves data from a bucket, encrypts it using a public key, and places it in another bucket. A user can then access that bucket through SharePoint, download the data, and decrypt it using a private key. This automates a process that was previously done manually and makes it a lot more secure with the use of encryption. The pipeline created means that developers can constantly make incremental changes to this code safely.

##### Knowledge

K20 - Pair/mob programming techniques and when to use each technique.

K23 - The importance of continual improvement within a blameless culture.

##### Skill

S8 - Work in agile, multi-disciplinary delivery teams, taking a flexible, collaborative and pragmatic approach to delivering tasks.

S13 - Engage in productive pair/mob programming.

##### Behaviour

B1 - Exhibits enthusiasm, openness and an aptitude for working as part of a collaborative community; e.g. sharing best practice, pairing with team members, learning from others and engaging in peer review practices.

Project #2

##### Description

The project is a large job that will affect multiple teams within the SCE Directorate. Within the directorate there are multiple silos of technical stacks, for example Notifications, and ECMP, etc. with multiple teams working within them. The task would be to create an accessible single view location across the entire directorate. This location would have dashboards covering various aspects and display monitoring for all the projects, with all issues highlighted. These issues would then be triaged effectively by the respective team. Effectively this would mean that any issue can be compared across all the projects and give visibility to whether it was a lone issue or affecting multiple projects. It would be a huge update and improvement on a system already in place that does not house the monitoring in one place.

##### Process

Workshop with Tom Farrow  
Tyler and I went into the Manchester office for a workshop with the DevOps lead for our team, Tom. It being the first time I’d been in the building to work, Tom led us through the basic health and safety procedures, including fire evacuation. We then went through the requirements for the project and came up with an idea of the architecture required to get it to work. The basic structure would be a User accessing Grafana to query Prometheus and Alert Manager. There are of course many restrictions on what we can and cannot do, however there’s also a few different ways to achieve the goal. I suggested an easy way to handle the load would be via a load balancer, which you can see in the diagrams (figs. 9 and 10). In the workshop we worked on a diagram, and Tyler digitised the diagram to this that displays our initial attempt at the architecture required to continue with the project.   
[K19](#Knowledge_2), [S21](#Skill_2)

*Text, whiteboard

Description automatically generated*

Figure 9 - Initial whiteboard sketch

*Diagram

Description automatically generated*

Figure 10 - Initial Diagram after Workshop

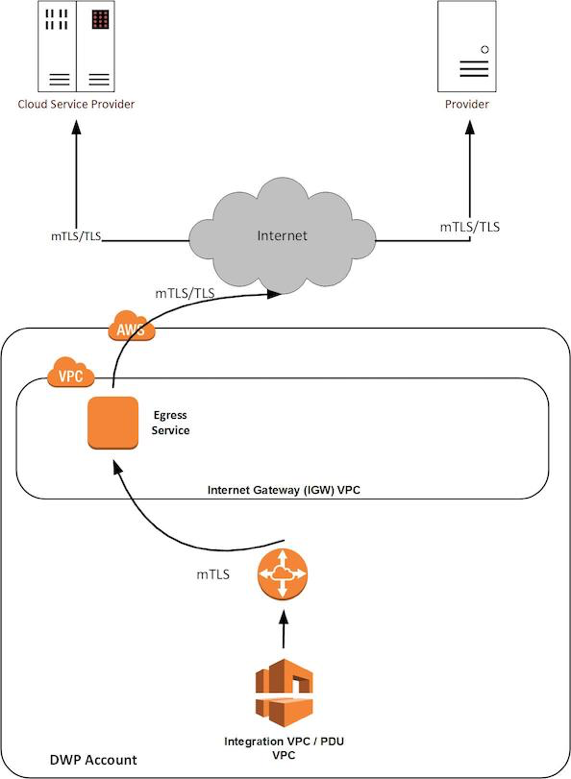
  
Architecture Call with Matthew Mann  
Following our workshop with Tom, we were told we had to run the architecture design past Matthew Mann, the architect for the Notifications project. Matthew accessed our proposal for compatibility with DWP data protection and highlighted the importance of keeping citizen information secure to comply with GDPR. He advised us to use a different pattern (fig. 11) instead of the transit gateway that we pitched, and he linked us to the correct pattern.   
[K3](#Knowledge_2), [S8](#Skill_2)

Figure 11 - Internet Egress Pattern provided by Architect

End of the Project  
The project then was put on hold, as our Line Manager had been working to find some way to outsource the whole project, and finally found a solution that would require minimal work from us. It was an interesting learning journey, however and I took a lot from it. This cut down a lot of man-hours and freed up the team to focus on other more pressing tickets and bugs, enforcing the DevOps ideal of not wasting time and money. The solution is a program called Dynatrace. Dynatrace uses AI and automation to monitor applications, container orchestration services, and infrastructure to provide observability of the full stack. This is effectively exactly what Tyler and I were planning on creating, and once our line manager had found this software there was a question of whether he was setting us off to try and re-invent the wheel. Although it would have been an interesting learning experience for both of us, it would have taken time, manpower, and money to eventually provide a working solution that would hopefully be able to do half of what Dynatrace can do. Given that it was useful to implement immediately, the quicker and easier option was taken, and not even at the expense of quality.   
[K24](#Knowledge_2)

|  |  |
| --- | --- |
| Pros of Dynatrace implementation | Cons of Dynatrace implementation |
| More capability (AI means it can learn over time, much more powerful than Prometheus/Grafana) | Less experience for Tyler and myself |
| Faster rollout | Potentially more expensive (depending on whether up-front cost is greater than the hours it would have taken the team) |
| More intuitive interface | External software not managed by us, potentially making it more difficult to troubleshoot if things go wrong |
| Less input needed from team |  |
| Out of the box software – up to date, does what it says on the tin, comes with a helpline |  |

##### Knowledge

K3 - How to use data ethically and the implications for wider society, with respect to the use of data, automation and artificial intelligence within the context of relevant data protection policy and legislation.

K19 - Different methods of communication and choosing the appropriate one - e.g. face-to-face (synchronous, high bandwidth), instant messaging, email (asynchronous, low bandwidth), visualisations vs. words.

K24 - The difference between Software-as-a-Service (SaaS) v bespoke v enterprise tooling and how to make an informed choice that suits each use case.

##### Skill

S8 - Work in agile, multi-disciplinary delivery teams, taking a flexible, collaborative and pragmatic approach to delivering tasks.

S21 - Application of lightweight modelling techniques, such as whiteboarding, in order to gain consensus as a team on evolving architecture.

##### Behaviour

Project #3

##### Ticket Name

NSE-2483

##### Ticket Description

After deploying the TMA-APP the pipeline should refresh the ASG so the new version of the APP would become available without further intervention. Graphical user interface, text, application, email

Description automatically generated

Figure 12 - Ticket screenshot

The pipelines created by these repos are used by the DevOps engineers to automate a lot of tasks, give visibility into what is happening, and help us to see what has and hasn’t been deployed. They’re an integral part of a continuous integration and continuous deployment method. However, we’re not the only people to use these pipelines. Developers utilise them when merging or deploying code to see what has and hasn’t failed, and they are useful for the testers too, as the pipeline runs tests for errors in code and makes sure it is up to regulatory requirements. The pipeline enforces various rules such as coverage, compliance, security, and analysis. It measures code coverage, linting, and other useful features, and throws a warning or error when something is not as expected. A lot of the data processed by the projects I work on is personal and sensitive data for DWP users, and so there are stringent rules around data protection. We make sure our work complies with the ITHC (IT Health Check), and there are various tools (such as SonarQube for code scanning) that we incorporate into the CI/CD pipelines to do vulnerability scanning within the packages and base images we use to make sure they comply to ITHC standards. On any update to these regulations, it is vital that we stay on top of sensitive data, and CI/CD pipelines enable us to update code quickly and seamlessly.   
[K22](#Knowledge_3)

##### Ticket Process

Meeting with Tom  
We went through what the ticket was asking, and Tom outlined the idea of creating a new step in the Gitlab CI pipeline which would manually refresh the Auto-Scaling groups (fig. 13). There were a few ways to successfully complete this task, but this seemed to me the most useful, and efficient way of creating a reusable piece of infrastructure. The idea to make the refresh a manual button was driven from a user experience standpoint, making it into an easy task executable by anyone.

Graphical user interface, text, application

Description automatically generated

Figure 13 - The desired result, showing completed manual 'refresh' jobs

Creating the script  
I wrote the initial script using some documentation that Tom had suggested I look at. The script had to do three major things: access the correct environment; assume the correct role; start the instance refresh. Having written that I hit various errors, that made me realise I’d have to install pip3 and the AWS CLI in the script as well.

Creating the fragment  
Once the script was working within the pipeline, refreshing the DSE1 ASG as expected, I could then get to work extracting the code to a fragment. The first thing I did was message my colleague Luis with my script, and some questions about fragments. He reviewed my code and suggested some changes, and other elements I might want to include in the process. He also linked some documentation and example repos with fragments.

I went away and did some learning, and then set up a call with myself and some other apprentices that I knew had worked with some fragments before (fig. 14). This was helpful, and we discussed the different uses of fragments between our teams. No one there had actually created one themselves yet, but after the knowledge sharing, I felt more confident to go and experiment myself.   
[S4](#Skill_3), [B4](#Behaviour_3)Text

Description automatically generated

Figure 14 - Setting up a call with apprenticeship colleagues to share knowledge

I extracted the versatile code to a hidden job at the start of the GitLab CI file and used an ‘include’ statement in the ‘asg-refresh’ job I had set up. I then took out as much code from the refresh job to the hidden job, using variables where appropriate. Once I’d done that, I moved the hidden job to a file in another repo, to store it with other fragments. I then included the file by using the filepath at the start of the CI file, and extended the fragment in my original ‘refresh’ job. I tested the script still worked, then opened a merge request for both new additions.

Show and Tell  
After my project was finished I reached out to the same apprentices who I’d called earlier in the week. I put together a quick presentation about the process that I’d taken and walked through it with them. I explained my process start to finish and took questions at the end. I explained to them how our pipelines are configured and talked through our pipeline file, before showing what my extra pipeline stage did, and how it added value to the team. Because they did not have experience in this method of extracting code to a fragment in a pipeline, they had a lot of questions about the use of variables and how that enabled the code to be as versatile as it was. I then presented the same thing to my DevOps team in the next morning stand-up. They had a much larger knowledge base than my fellow apprentices and understood the logic behind the actual fragment. What I spent more time explaining to them was the benefit this would have in our day-to-day work schedule, and how this might impact other repos. I could talk at a much higher level about the impact this might have on the business, as opposed to being very clear on the logic in the actual code, which, although in essence was the same presentation, meant I was focussing on completely different aspects of the project based on the understand of the general audience.   
[S1](#Skill_3)

Business Impact  
The fragment I created enables DevOps engineers to refresh the Auto-Scaling Groups manually, without having to redeploy the whole pipeline every time a minor change is made. It also means that SRE engineers don’t have to be involved as they normally would, making the whole workflow smoother (fig. 15). This embodies the DevOps value of building on internal quality, reducing waste and effort. The reason we wanted this change added was to reduce reliance on SRE. Originally we would collaborate with SRE in this matter, as we as DevOps engineers lack the permissions to refresh ASGs, however due to the difference in in frameworks used it became clear that it would be much smoother to reduce this dependency. Often the SRE contacts we have are working on different schedules to us, or are busy with more high priority tasks, as they are required to provide support for some systems around the clock, 24/7. This makes communication with SRE more asynchronous and can occasionally lead to waiting times or a slower turnaround on tickets, as we may have to wait for SRE to do a small blocking task before we can move forward. This fragment I introduced removes that small blocker and makes workflow much smoother.   
[S2](#Skill_3)  
A picture containing text

Description automatically generated

Figure 15 - a fellow DevOps engineer letting me know when he used my solution!

##### knowledge

K22 - How their occupation fits into the wider digital landscape and any current or future regulatory requirements.

##### Skill

S1 - Communicate credibly with technical and non-technical people at all levels, using a range of methods; e.g. ‘Show and Tell’ and ‘Demonstrations’.

S2 - Work within different organisational cultures with both internal and external parties.

S4 - Initiate and facilitate knowledge sharing and technical collaboration

##### Behaviour

B4 - Is inclusive, professional and maintains a blameless culture.

Project #4

##### Ticket Name

DDMDEV-147

##### Ticket Description

This was a quick ticket that my line manager asked for some help on. It was part of the implementation of DynaTrace. In the IAC scripts, DynaTrace is installed on the EC2 instances. The script being used in one of the images was this:

if [ $ENVIRONMENT == 'DSE2' ]

then

wget --no-check-certificate -O Dynatrace-OneAgent-Linux.sh "URL" --header="Authorization: Api-Token ab0c01.ABCDEFGH1HI23JK45YLMNOPQRS.HUG3OO7YUFKRGQATWJGTTIBATHDLYGNIYI2MHEWJQ2MFG3VKL4QJ7UAGSSCIDD32"

/bin/sh Dynatrace-OneAgent-Linux.sh --set-host-group=PRE\_TMA\_${ENVIRONMENT}

fi

This checks if the environment is DSE2, and if so it downloads DynaTrace, installs it, and sets the host-group to PRE\_TMA\_<env>. Tom needed the script changed so that:

1. It follows all those steps unless the environment is PRD or PTE
2. The host-group name becomes PRE-NOTS-<env>
3. In the environments PRD or PTE it should do a similar thing but have a different URL
4. In PRD and PTE the host-group name becomes PRD-NOTS-<env>

This ticket follows a plan-do-check-act scheme. The PLAN stage takes place when I get on a call with my LM and simply talk through the brief, what is required, and how I might go about completing it. Then the DO stage, when I go away and write up the code that is required. The CHECK stage happens when I reconvene with my LM, explain to him what I am unsure of in the code, and he proposes a simple way I can check it. Then the ACT stage occurs when the script I have written is approved, and I am able to merge it so it becomes a functioning and useful part of our codebase.   
[K6](#Knowledge_4)

##### Ticket Process

Call with Tom  
On the call with Tom he outlined all the above information to me and explained more about DynaTrace and the way it works. Initially this job would have been to install DynaTrace on all the Lambdas too if there were any that Tom had missed, however after having checked them I realised he’d been very thorough. It was useful to talk through a plan of action, even for something as simple as this. I explained a basic plan of a script to use, in simple terms.

Proposing a script  
I quickly got back to Tom after the call with a script that looked like this:

if [ $ENVIRONMENT == 'PTE\*' ] || [ $ENVIRONMENT == 'PRD\*' ]

then

wget --no-check-certificate -O Dynatrace-OneAgent-Linux.sh "[URL](https://activegate.nonprod.dwpcloud.uk:9999/e/78c2cf55-f339-477e-bd2f-4d24a3071baf/api/v1/deployment/installer/agent/unix/default/latest?arch=x86&flavor=default)" --header="Authorization: Api-Token ab0c01.ABCDEFGH1HI23JK45YLMNOPQRS.HUG3OO7YUFKRGQATWJGTTIBATHDLYGNIYI2MHEWJQ2MFG3VKL4QJ7UAGSSCIDD32"

/bin/sh Dynatrace-OneAgent-Linux.sh --set-host-group=PRD\_NOTS\_${ENVIRONMENT}

else

wget --no-check-certificate -O Dynatrace-OneAgent-Linux.sh "ALT\_URL --header="Authorization: Api-Token ab0c01.ABCDEFGH1HI23JK45YLMNOPQRS.HUG3OO7YUFKRGQATWJGTTIBATHDLYGNIYI2MHEWJQ2MFG3VKL4QJ7UAGSSCIDD32"

/bin/sh Dynatrace-OneAgent-Linux.sh --set-host-group=PRE\_NOTS\_${ENVIRONMENT}

fi

A simple ‘if’ statement that checked if the environment was PTE or PRD and did a different thing depending on the outcome. There were a few things in this script I was unsure about. The key thing being the ‘wildcards’ I had added after the envs PTE and PRD. I added this to ensure that we could use the same script for DSE1 and DSE2, however I was unsure of the best way to check if the wildcards worked the way I thought they did – beyond googling it (fig. 16)

Text

Description automatically generated

Figure 16 - Proposing my script to Tom, and setting up a meeting to check it

Checking the script  
I hopped back on a call with Tom and he suggested a quick and easy way to check the wildcards. Using my console I wrote a bash script that did pretty much what I was attempting, however much simpler (fig. 17). A screenshot of a computer

Description automatically generated with medium confidence

Figure 17 - A recreation of the tests I ran in the console

I found out that the wildcards didn’t work in bash the way I was expecting. Instead, it seemed to produce all the files in a directory. Wildcards only worked the way I was remembering them in regex, and a double square bracket converts bash to regex. I could then test if they worked and applied this knowledge to the original script.

Applying the script  
I made the changes to the script, pushed it to a feature branch, and requested review from Tom. He approved and I merged it into DSE2 (fig. 18).

Graphical user interface, text, application

Description automatically generated

Figure 18 - Creating the MR and merging to DSE2

Business impact  
DynaTrace is a hugely valuable piece of software that handles monitoring and logs for our entire stack. Getting this up and running quickly is of very high priority, so additions like this script are key to getting this process underway. This will improve efficiency in triaging errors, reduce time searching for bugs, and potentially warn us of problems before they happen.

##### Knowledge

K6 - A range of problem-solving techniques appropriate to the task at hand, such as affinity mapping, impact maps, plan-do-check-act/Deming.

Project #5

##### Ticket Name

DDMDEV-184

##### Ticket Description

Write the terraform for a new ApiGateway and Lambda and deploy into the Developer Supported Environment.

Graphical user interface, text, application

Description automatically generated

Figure 19 - The ticket I was assigned

##### Ticket Process

Picking up the ticket  
Hot off the back of some self-development, I joined a morning stand-up and was chosen to pick up a ticket from one of the other members of the team who had to transition to something more urgent. Graphical user interface, text, application

Description automatically generated

Figure 21 - Tom messaged me after the stand-up with a link to the ticket and repo

Working on the ticket  
This was an interesting ticket for me as it was the first time that I was effectively solo on a major ticket and interacting directly with developers by myself. My first port of call was to message Pavan (fig. 22) and try and understand how far he’d got with the work. He’d added some basic skeleton code into the repo, but it was somewhere to start for me. Graphical user interface, text

Description automatically generated

Figure 22 - Messaging Pavan to check what he had completed for the ticket

I then messaged Paul (Shridhar was off that day), to have a quick look at what else was needed in the repo, and he gave me some helpful feedback (fig 23). Text

Description automatically generated

Figure 23 - Paul took the time to write some helpful comments

I started addressing the changes Paul made and made a few pushes onto the feature branch. When Shridhar was back, I connected with him (fig. 24) and made some initial changes, merging in some changes he had made. Text

Description automatically generated

Figure 24 - Connecting with Shridhar and explaining where I was up to

He messaged me saying he was receiving this error:

{

“level”: “error”,  
 “time”: “2022-08-17T14:31:08.433Z”,  
 “name”: “guid-adapter\_handler”,  
 “message”: “Something went wrong: {\“code\“:500,\“message\“:\“Error while retrieving parameter info for Parameter Name = https://production-exchange-platform-idp.auth.eu-west-2.amazoncognito.com and reason: AccessDeniedException: User: arn:aws:sts::1234567789:assumed-role/NOTS-DSE2-notificationsService-GUIDAdapter-Handler/NOTS-DSE2-notificationsService-GUIDAdapter-Handler is not authorized to perform: ssm:GetParameter on resource: arn:aws:ssm:eu-west-2:1234456789:parameter/https because no identity-based policy allows the ssm:GetParameter action\“}”

}

Which I fixed, along with a few other bugs and fixes that needed to be sorted from a DevOps side. Text

Description automatically generated

Figure 25 - Updating Shridhar with the fix I had implemented

It was interesting having a bit more of an insight into the different roles within the larger team, understanding the channels of communication, and who required what from whom. For example, in this ticket I communicated with Pavan first, one of the DevOps engineers from the Notifications team. I then connected with Paul, our Application Architect to get further insight into the component and what it was trying to do. As lead architect Paul works taking the requirements we need to implement and breaking them down to the correct level for us to tackle them. I also worked with Shridhar, the Lead Developer for this project, whose job it is to program and build the software. On completion of the project I updated some documentation at the request of Rachel, our Team Manager, who takes care of the timelines, resource management and other requirements. It helped put it into a clearer view of who does what, and why.  
[K18](#Knowledge_5)

Handover to Aravind  
After completing most the IAC needed for the ticket, I handed it over to a fellow DevOps engineer Aravind. This was because the ticket was suddenly upgraded to a high priority ticket, and although it was a great learning experience for me to interact with different teams and people, there was call for some fast troubleshooting. It made more sense for me to continue learning, consolidate what I had learned throughout the process of the ticket, and shadow Aravind in his helping of the devs, rather than trying to answer questions from the devs that would take me a few hours to understand.

Business Impact  
The GUID Adaptor is a small internal micro-service that acts as a centralised pass-through service for any internal Nots service that needs to call the DWP GUID service. It abstracts the Cognito authentication token process required to access the GUID service away from the consuming Nots services, constructing the access token required for authentication and injecting this as a header to any requests before passing them onto the GUID service itself; it also handles recycling the Cognito authentication token which only has a lifespan of ~1hr. Any response from the GUID service will then be relayed directly back to the internal Nots consuming service.

##### Knowledge

K18 - Roles within a multidisciplinary team and the interfaces with other areas of an organisation.

Project #6

##### Description

Personal Professional Development.

##### process

Scrum training  
Just before picking up this ticket I completed the Professional Scrum Master I exam. I chose to do this at this time because there was a slight lull in my workload, so I set aside a day to revise and complete the exam. I was pleased to pass first time (fig. 20). It became apparent to me that sitting the exam would be a valuable use of my time, and I was keen to invest some extra time into gaining new knowledge and achievements, developing the skills I had learned over my apprenticeship. I chose the PSM I exam, as it was recommended to me after a training course we did as part of the apprenticeship. Given that my team works with an agile mindset, it made sense to make sure I understood Scrum as being one of the key methodologies that applies agile. Completing this course made me realise I wanted to further my learning in other areas, and that it would be a good idea to widen my base of knowledge further with more training. I spoke to my supervisor, and she advised me that Careers Toolbox is a tool that will soon be available to me that will help with keeping track of what I know, asses my skill level against my role, create action plans, and track my learning. This sounds incredibly helpful, and I can’t wait to implement this into my professional development. Taking the Scrum course had an impact on my day-to-day life, as I suggested to my Line Manager to implement some of the concepts of Scrum into our daily standups. One of the requirements from the Scrum Master is removing impediments to the developing team’s progress, and, while my team is not strictly following Scrum methodology, it was useful to implement even a small thing like Tom checking in with everyone to ask if they had faced any blockers in the previous day. This has made our standups much more useful and engaging, as I get to hear different DevOps engineers tackle problems together and come to solutions.  
[S16](#Skill_6), B2

Graphical user interface, text, application, email

Description automatically generated

Figure 25 - Michael Plasom-Scott, Professional Scrum Master

AWS Certifications  
I am currently working towards completing a course on A Cloud Guru on Packer. Packer is a lightweight tool for building machine images on AWS, as well as a range of other platforms. I’m hoping to use this knowledge of Packer in my upcoming final apprenticeship project. Resources like A Cloud Guru and Pluralsight enable me to stay aware of learning resources available to me. The APS I qualification I gained, and my attendance of the week long Scrum training helped me to understand the quality of certifications out there. They could help me to better my understanding and help me to become more knowledgeable and efficient in my work. I also have an AWS certification account, and have been made aware of various AWS certifications that would be incredibly insightful to me. I am particularly keen to look into a workshop AWS have available on Blue/Green and Canary deployments, as this is what I may be focusing on for my final project of the apprenticeship. This is a great option if I ever want to specialise as well, as AWS offer speciality options. Amazon has a high standard for their pass rate, so taking these courses would enable me to be confident in my understand, which is why they are so valuable to my learning.   
[K25](#Knowledge_6)

##### knowledge

K25 - Maintain an awareness of cloud certification requirements.

##### skill

S16 - Invest in continuous learning, both your own development and others, ensuring learning activities dovetail with changing job requirements. Keep up with cutting edge.

##### behaviour

B2 - Invests time and effort in their own development, recognising that technology evolves at a rapid rate.

## KSBs

##### Knowledge

K3 How to use data ethically and the implications for wider society, with respect to the use of data, automation and artificial intelligence within the context of relevant data protection policy and legislation.

K6 A range of problem solving techniques appropriate to the task at hand, such as affinity mapping, impact maps, plan-do-check-act/Deming.

K9 Different organisational cultures, the development frameworks utilised and how they can both complement each other and introduce constraints on delivery.

K18 Roles within a multidisciplinary team and the interfaces with other areas of an organisation.

K19 Different methods of communication and choosing the appropriate one - e.g. face-to-face (synchronous, high bandwidth), instant messaging, email (asynchronous, low bandwidth), visualisations vs. words.

K20 Pair/mob programming techniques and when to use each technique.

K22 How their occupation fits into the wider digital landscape and any current or future regulatory requirements.

K23 The importance of continual improvement within a blameless culture.

K24 The difference between Software-as-a-Service (SaaS) v bespoke v enterprise tooling and how to make an informed choice that suits each use case.

K25 Maintain an awareness of cloud certification requirements.

##### Skill

S1 Communicate credibly with technical and non-technical people at all levels, using a range of methods; e.g. ‘Show and Tell’ and ‘Demonstrations’.

S2 Work within different organisational cultures with both internal and external parties

S4 Initiate and facilitate knowledge sharing and technical collaboration

S8 Work in agile, multi-disciplinary delivery teams, taking a flexible, collaborative and pragmatic approach to delivering tasks.

S13 Engage in productive pair/mob programming.

S16 Invest in continuous learning, both your own development and others, ensuring learning activities dovetail with changing job requirements. Keep up with cutting edge.

S21 Application of lightweight modelling techniques, such as whiteboarding, in order to gain consensus as a team on evolving architecture.

##### Behaviour

B1 Exhibits enthusiasm, openness and an aptitude for working as part of a collaborative community; e.g. sharing best practice, pairing with team members, learning from others and engaging in peer review practices.

B2 Invests time and effort in their own development, recognising that technology evolves at a rapid rate.

B4 Is inclusive, professional and maintains a blameless culture.